

The author of the doctoral dissertation: Olga Springer  
Scientific discipline: Informatics

## **DOCTORAL DISSERTATION**

Title of doctoral dissertation: Software Product Management - challenges, guidelines, methods and techniques to improve software engineering practices.

Title of doctoral dissertation (in Polish): Zarządzanie produktem informatycznym - wyzwania, wytyczne, metody i techniki do poprawy praktyk inżynierii oprogramowania.

Supervisor	Auxiliary supervisor
<i>signature</i>	<i>signature</i>
dr hab. inż. Agnieszka Landowska	dr inż. Jakub Miler

Gdańsk, 2024



## **OŚWIADCZENIE**

Autor rozprawy doktorskiej: Olga Springer

Ja, niżej podpisany(a), oświadczam, iż jestem świadomy(a), że zgodnie z przepisem art. 27 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2021 poz. 1062), uczelnia może korzystać z mojej rozprawy doktorskiej zatytułowanej:

“Zarządzanie produktem informatycznym - wyzwania, wytyczne, metody i techniki do poprawy praktyk inżynierii oprogramowania.”

do prowadzenia badań naukowych lub w celach dydaktycznych.<sup>1</sup>

Świadomy(a) odpowiedzialności karnej z tytułu naruszenia przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych i konsekwencji dyscyplinarnych określonych w ustawie Prawo o szkolnictwie wyższym i nauce (Dz.U.2021.478 t.j.), a także odpowiedzialności cywilno-prawnej oświadczam, że przedkładana rozprawa doktorska została napisana przeze mnie samodzielnie.

Oświadczam, że treść rozprawy opracowana została na podstawie wyników badań prowadzonych pod kierunkiem i w ścisłej współpracy z promotorem dr hab. inż. Agnieszką Landowską oraz promotorem pomocniczym dr inż. Jakubem Milerem.

Niniejsza rozprawa doktorska nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem stopnia doktora.

Wszystkie informacje umieszczone w ww. rozprawie uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami, zgodnie z przepisem art. 34 ustawy o prawie autorskim i prawach pokrewnych.

Potwierdzam zgodność niniejszej wersji pracy doktorskiej z załączoną wersją elektroniczną.

Gdańsk, dnia .....

.....  
*podpis doktoranta*

Ja, niżej podpisany(a), wyrażam zgodę/nie wyrażam zgody\* na umieszczenie ww. rozprawy doktorskiej w wersji elektronicznej w otwartym, cyfrowym repozytorium instytucjonalnym Politechniki Gdańskiej.

Gdańsk, dnia .....

.....  
*podpis doktorant*

<sup>1</sup> Art. 27. 1. Instytucje oświatowe oraz podmioty, o których mowa w art. 7 ust. 1 pkt 1, 2 i 4–8 ustawy z dnia 20 lipca 2018 r. – Prawo o szkolnictwie wyższym i nauce, mogą na potrzeby zilustrowania treści przekazywanych w celach dydaktycznych lub w celu prowadzenia działalności naukowej korzystać z rozpowszechnionych utworów w oryginale i w tłumaczeniu oraz zwielokrotniać w tym celu rozpowszechnione drobne utwory lub fragmenty większych utworów.

2. W przypadku publicznego udostępniania utworów w taki sposób, aby każdy mógł mieć do nich dostęp w miejscu i czasie przez siebie wybranym korzystanie, o którym mowa w ust. 1, jest dozwolone wyłącznie dla ograniczonego kręgu osób uczących się, nauczających lub prowadzących badania naukowe, zidentyfikowanych przez podmioty wymienione w ust. 1.

## TABLE OF CONTENTS

Table of contents .....	3
Acknowledgements .....	6
Abstract .....	7
1. Introduction .....	8
1.1. Motivation .....	8
1.2. Objectives and Limitations of scope .....	9
1.3. Dissertation hypothesis .....	9
1.4. List of acronyms .....	9
1.5. Layout .....	10
2. Literature review .....	11
2.1. Specificity of <b>software</b> product management .....	11
2.2. Related research .....	13
2.2.1. Product Management Frameworks .....	16
2.2.2. Problems related to software product management [13] .....	17
2.2.3. Strategies to software product management [20] [3] [13] .....	18
2.3. Research gaps .....	20
3. The proposed Software Product Management Guide .....	21
3.1. Components of the proposed framework .....	21
3.2. Who can use the framework and what are the benefits? .....	23
3.3. Framework application as a diagnostic tool .....	23
3.4. Framework description - problems component .....	24
3.5. Framework description - Solutions, Guidelines and Techniques components .....	27
3.5.1. Problem: Determining the true value of the product that the customer needs (P74) .....	27
3.5.2. Problem: Strategy and priorities are changing frequently (P35) .....	30
3.5.3. Problem: Technical debts (P9) .....	32
3.5.4. Problem: Working in silos (problem with communication, synchronisation between teams) (P64) .....	35
3.5.5. Problem: Balancing between reactive and proactive work (P69) .....	37
3.6. Framework - Supporting Techniques .....	41
3.7. Chapter summary .....	44
4. Studium 1 - Problems related to software product management [19] .....	45
4.1. Systematic literature review .....	46
4.2. Interviews .....	53
4.3. Survey .....	59
4.4. Summary .....	64

5. Studium 2 - Strategies to main problems related to software product management .....	65
5.1. Study design .....	65
5.2. Focus groups study .....	65
5.2.1. Planning the Research .....	65
5.2.2. Designing focus groups .....	67
5.2.3. Conducting the focus group sessions .....	68
5.3. Validating solutions with Survey .....	69
5.4. Creating a list of supporting techniques .....	71
5.5. Summary .....	71
6. Studium 3 - SPMG Framework validation .....	72
6.1. Study concept .....	72
6.2. Case study decisions .....	73
6.3. Framework evaluation activities .....	75
6.4. Case study process .....	76
6.4.1. Preparation stage .....	76
6.4.2. During the session .....	77
6.4.3. Reporting and Documentation Stage .....	78
6.4.4. Gathering feedback .....	78
6.5. Participants of case study .....	78
6.6. Case study: Team A .....	79
6.6.1. Step 1: Analyse context - general survey to get familiar with the team, their project and participants' roles .....	79
6.6.2. Step 2: Diagnose product management state - identify key problems .....	80
6.6.3. Step 3: Select strategies - guidelines, solutions and techniques for one of the most significant problems that the team identified .....	81
6.6.4. Step 4: Summarise session .....	83
6.6.5. Step 5: Prepare a SPMG Audit Report .....	83
6.6.6. Framework evaluation .....	83
6.7. Case study: Team B .....	85
6.7.1. Step 1: Analyse context - general survey to get familiar with the team, their project and participants' roles .....	85
6.7.2. Step 2: Diagnose product management state - identify key problems .....	86
6.7.3. Step 3: Select strategies - guidelines, solutions and techniques for one of the most significant problems that the team identified .....	89
6.7.4. Step 4: Summarise session .....	91
6.7.5. Step 5: Prepare a SPMG Audit Report .....	91
6.7.6. Framework evaluation .....	92
6.8. Case study: Team C .....	93
6.8.1. Step 1: Analyse context - general survey to get familiar with the team, their project and participants' roles .....	93
6.8.2. Step 2: Diagnose product management state - identify key problems .....	94

6.8.3. Step 3: Select strategies - guidelines, solutions and techniques for one of the most significant problems that the team identified .....	94
6.8.4. Step 4: Summarise session .....	95
6.8.5. Step 5: Prepare a SPMG Audit Report .....	95
6.8.6. Framework evaluation .....	96
6.9. Case study: Team D .....	96
6.9.1. Step 1: Analyse context - general survey to get familiar with the team, their project and participants' roles .....	97
6.9.2. Step 2: Diagnose product management state - identify key problems .....	97
6.9.3. Step 3: Select strategies - guidelines, solutions and techniques for one of the most significant problems that the team identified .....	98
6.9.4. Step 4: Summarise session .....	100
6.9.5. Step 5: Prepare a SPMG Audit Report .....	100
6.9.6. Framework evaluation .....	101
6.10 Comparative analysis of results between teams .....	102
6.11 Thesis validation .....	104
6.11.1. Statistical analysis .....	105
6.11.2. Qualitative feedback on the framework .....	105
7. Summary .....	107
Bibliography .....	115
List of figures .....	116
List of tables .....	118
List of appendixes .....	119

## **ACKNOWLEDGEMENTS**

I would like to express my sincere thanks to dr hab. eng. Agnieszka Landowska for invaluable help provided during the preparation of the doctoral thesis, introduction to issues of software engineering practices - from the beginning of the second-degree studies, as well as continuation during the doctorate - over 8 years of cooperation. Special thanks for patience, dedicated time, and numerous conversations that were an inspiration and raising my spirits in moments of hesitation - it enabled me to complete the work.

Special thanks also for dr Jakub Miler - thanks for the inspiration to undertake a doctoral thesis and a guide in the first phases of the research.

Special thanks to my husband Igor for his patience, understanding and tremendous support and faith in me. If it weren't for you, I wouldn't have had the strength to complete the dissertation.

I would like to thank the entire Department of Software Engineering for their support and accurate comments during seminar presentations and behind-the-scenes talks.

I would also like to thank my family and all my friends from whom I received and felt words of encouragement. Thank you for believing in me and your support.

## **ABSTRACT**

Software product management, which involves multiple processes and responsibilities, and links to many activities within the organisation, increases the success rate of IT projects.

Nowadays, in the scientific literature there is no source of knowledge about the most frequent and relevant problems related to software product management and what are the potential strategies to solve them. It was not analysed yet if practices used by software product managers can positively influence software engineering practices used by software teams.

The main goals of this PhD thesis were to propose and validate a new framework that lead teams to diagnose their software product management challenges and propose a solution that can be applied in their environment.

After reviewing the literature and identifying research gaps, a series of studies were planned to define the main problems related to software product management and potential solutions to these problems. In the research, problems were identified through the use of the systematic literature review, interviews and a survey, while the focus group technique was used to identify solutions, guidelines and supportive techniques. Then case study research was performed to evaluate the framework. All together, over 170 software product managers were involved in the research to form and evaluate the proposed framework.

The main achievement of this thesis - namely Software Product Management Guide constitutes a comprehensive set of problems, guidelines, solutions and techniques to evaluate and improve software engineering practices with regard to software product management. The proposed framework help product development teams to optimise their software engineering practices. The proposed quantification method allows for a numerical representation of how teams perceive problems in their organisations and how they rank potential solutions.

Keywords: software product management, software engineering, software product manager, software product development

# 1. INTRODUCTION

## 1.1. *Motivation*

Software and product terms are different. The product is a higher order entity. Software may be a component of a more compound product or a service, but it can be also treated as the product itself. The term software product management emerged in 1997, marking the beginning of its evolution [1]. Companies that produce software started to offer it as a service or as a product. This is why software product management is becoming increasingly recognized in software development companies.

There have already been many successful organizations implementing software product management (such as Microsoft, IBM, and Google). The role of the software product manager is an important factor in the economic success of the product and the company investing in it [2]. It has been shown that the institutionalization of this role measurably improves the success rate of projects [3]. Software product management involves multiple processes and responsibilities and links to many activities within the organisation. However, the adoption of software product management activities and the implementation of the software product manager role itself varies between companies.

Software product management is a set of processes aimed at defining, introducing, developing, growing, maintaining and withdrawing a software product on the market. It is closely related to other areas of software engineering such as strategy development, requirements engineering, project management, agile software development, product marketing, and business analysis [4] [5].

Software product management got integrated into the commonly used software development and management frameworks in the IT industry [6] [7]. Software product management diverges from project management by placing greater emphasis on customers, sales, user feedback, and continuous product growth. Products are typically developed without strict time constraints, by involving multiple projects within a broader timescale.

Scrum supports product management as a framework, as it allows for continuously improving the product, the team, and the working environment. Scrum Guide introduces the role of the product owner, which is responsible for business goals and value, although it does not specify precisely how this role should be performed effectively [6]. Project management methodologies and guidelines such as PRINCE2® and PMBoK focus on project delivery and techniques for project managers, rather than on how to create a valuable product for customers [5] [8].

The problems and risks related to project management, requirements engineering, Agile, or UX are already widely discussed in the literature [9] [10] [11] [12]. For software product management specifically, the one of just a few articles focused on the software product management problems is entitled “Lean Solutions to Software Product Management Problems” by Maglyas et al. [13]. They compiled a short list of 5 quite generic problems without any information about their occurrence or scale of impact on product management. Product management is gaining popularity every year, however it has been topic for scientific research only few times in the recent years.

This indicates that the research on the topic of the problems of software product management is still open, as the understanding of the specific problems and the evaluation of their frequency



and severity require further studies [14] [15] [16] [17] [18] in the literature. It was pointed out that it is crucial to understand the problems of software product management more deeply than specified by Maglyas. The same applies to strategies and techniques for the problems related to software product management. As problems were not well studied, it is hard to find appropriate solutions that will improve the software product management process and software engineering practices.

Identified research gaps:

1. Unknown specific problems related to **software** product management.
2. Lack of **structured strategies** for key software product management problems.
3. Lack of understanding if software product management practices used by product managers can positively **influence software engineering** practices in product development teams.

## **1.2. Objectives and Limitations of scope**

The goal of this research is to identify key problems related to software product management and define the solutions for these problems as well as validate if these strategies can influence software engineering practices in product development teams.

The goal can be decomposed as follows:

- propose a method to diagnose and prioritise software product management-related problems in software product development teams;
- propose a repository of guidelines, solutions and techniques that may be used to address main problems related to software product management;
- analyse if the proposed the solutions can positively influence software engineering practices in product development teams.

The following aspect is beyond the scope of this research: **products other than software**.

The main achievement of this thesis - namely Software Product Management Guide constitutes a comprehensive set of problems, guidelines, solutions and techniques to evaluate and improve software engineering practices with regard to software product management. The proposed quantification method allows for a numerical representation of how teams perceive problems in their organisations and how they rank potential solutions.

## **1.3. Dissertation hypothesis**

The thesis of this dissertation was given as follows:

***The proposed Software Product Management Guide framework influences positively software engineering practices in software product development teams.***

## **1.4. List of acronyms**

SPM - software product management

SPD - software product development

PM - product manager

SPMG - Software Product Management Guide

SLR - systematic literature review

## **1.5. Layout**

The second chapter provides an overview of the current landscape in software product management, offering insights into established frameworks that define the discipline and delineate the responsibilities of software product managers. Additionally, this chapter describes state-of-the-art in software product management, highlighting key studies that point out challenges associated with software product management, along with potential solutions.

In the third chapter, the author introduces a novel framework, providing description of its components and principles. Then, the chapters four to six present the studies that enabled to formulate and evaluate the proposed framework.

The fourth chapter presents Studium 1 aimed at identification and specification of software product management problems and its severity (influence on the software processes). The study described was published in Empirical Software Engineering [19]. The study allowed to define the first component (problems) of Software Product Management Guide framework - proposed in this thesis.

The fifth chapter introduces Studium 2, a research initiative designed to compile practical solutions and techniques employed by software product managers to address identified challenges. The study described was published in IEEE Access [20]. The study constitutes other components (solutions and techniques) of the framework proposed in this thesis.

The sixth chapter describes the validation process of the proposed Software Product Management Guide framework through collaboration with product development teams (Studium 3).

The final chapter encapsulates a synthesis of the achieved results and outlines potential future research and development of the proposed framework.

## 2. LITERATURE REVIEW

The chapter describes the state-of-the-art in the software product management and software development areas. The studies described were a starting point to the framework proposed in this thesis. The background terminology was also provided.

### 2.1. *Specificity of software product management*

In the realm of marketing, a **product** refers to an item, system, or service provided for consumer use in response to consumer demand. Essentially, it encompasses anything that can be presented to a market to meet the desires or needs of a customer. Within the retail sector, products are commonly termed as merchandise, while in manufacturing, raw materials are procured and transformed into finished goods for sale. Additionally, a service is considered a distinct category of product. [21].

Developing a **software product** differs significantly from developing a physical product. Building software products requires a lot of effort to design, develop, and test the solution. In many cases to cover the initial investment, the software product must be sold many times over. Unlike other products, software products do not require extra costs for manufacturing and distributing additional copies. Software products don't wear out but require maintenance. With software products that can be upgraded after the first sale, it is possible to offer customers new value in the future [22].

**Product management** involves the strategic business process of conceptualizing, creating, launching, and overseeing a product or service. This encompasses the entire journey of a product, from its initial idea through development to market introduction. Product managers play a crucial role in guaranteeing that a product aligns with the requirements of its target market, aligns with the business strategy, and is effectively managed across all stages of its lifecycle. In the software development, product management adapts the core principles of traditional product management to the unique characteristics of digital products. For example the role of product owner in some agile software development methodologies is the one applying principles of product management area.

Product management and software product management share common principles as presented in Figure 2.1, but they differ in focus and application. While product management is a comprehensive approach applicable to various industries and product types, software product management narrows its focus to the specific nuances of managing software products. The latter requires a more in-depth understanding of software development processes and technical considerations.

Software product management (sometimes also referred to as digital product management or, in the right context just product management) is the discipline of building, implementing and managing software or digital products, taking into account life cycle considerations and an audience. It is the discipline and business process that governs a software product from its inception to the market or customer delivery and service in order to maximize revenue. This is in contrast to software that is delivered in an ad hoc manner, typically to a limited clientele, e.g. service.

Software product managers play an important role in the software development organization

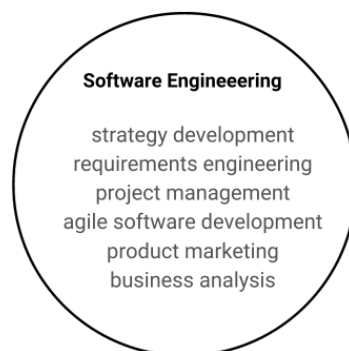
Product Management vs Software Product Management			
	Scope	Applicability	Skills
PRODUCT MANAGEMENT	Broader discipline that encompasses the end-to-end process of planning, developing, launching, and managing a product or service.	Applies to a wide range of industries and products, including physical goods, services, and software.	Product Managers need a mix of business, marketing, and project management skills to guide the overall product strategy.
SOFTWARE PRODUCT MANAGEMENT	Specifically deals with the planning, development, and management of software products.	Focused on software applications, platforms, or services, often in the context of technology companies or organizations with a significant software component.	Software Product Managers need a deep understanding of software development processes, technical requirements, and the unique challenges associated with software products.

**Figure 2.1:** Comparison of Product Management vs Software Product Management

while being responsible for the strategy, business case, product roadmap, high-level requirements, product deployment (release management), and retirement plan. By definition, product management is different from project management, however, in some software companies, the roles of product manager and project manager are mixed [2].

Software product management encompasses a range of processes that focus on the definition, introduction, development, growth, maintenance, and withdrawal of a software product from the market. It is intricately embedded in various domains within software engineering, including strategy development, requirements engineering, project management, agile software development, product marketing, and business analysis - as visualised in Figure 2.2 [4] [23].

Software engineering is typically a long, multi-step process that relies on a variety of team members—including those who identify what a product should include and those who design the product's software accordingly.

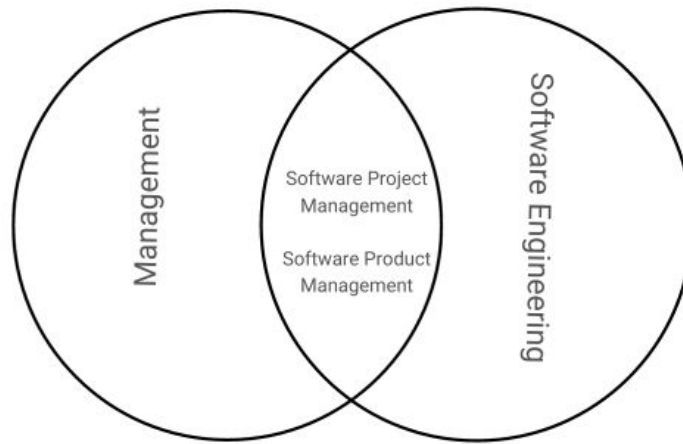


**Figure 2.2:** Examples of software engineering domains

While agile software development, which is a domain of software engineering, align with software product management in terms of their user-centred focus and swift delivery of business value, they do not address the higher-level strategic aspects of product development. The

business analysis covers an important part of product management, but it lacks a technical viewpoint. Software product management effectively spans all of these areas.

Relation between Management, Software Engineering and Software Product Management is presented in Figure 2.3.



**Figure 2.3:** Relation between management, software engineering and software product management

Software Product Managers do not only follow the product life cycle, they own it. They are responsible for the success of the product, which requires them to take care of wide technical and business activities: product strategy, product planning, strategic management, and orchestration of the organization's functional areas such as: product architecture management, development execution, users experience design, requirements engineering, quality management [2].

Scrum is the most popular framework, commonly used in software development and other industries. There is a role defined in Scrum that is responsible for requirements and business value – the product owner [6]. Spectrum of activities and responsibilities of the product manager is much wider than the product owner role [24].

The approach to software product management often follows the principles of lean management as in the Lean Start-up method [25]. Product managers design and verify the business models in the market with various techniques and tools [26].

## **2.2. Related research**

There is a study from 2010 that provides a structured assessment of software product management [27]. The authors propose the maturity matrix for software product management, concentrating on the functions. After developing the matrix, they used a survey in which 45 software product managers reported by indicating the right order in which software product management capabilities should be implemented in an organization. The survey enables product managers to benchmark their organization, assess individual processes and apply best practices to create an effective software product management environment. The main areas covered by the assessment survey are requirements gathering, system requirements identification, requirements organizing, requirements prioritization, software release definition, release definition validation, scope change management, launch preparation, core asset roadmapping, product roadmapping, roadmap intelligence, partnering and contracting, and product lifecycle

management. Since 2010 understanding and implementation of software product management evolved, so as of today defined software product management capabilities in this paper and the usage of the matrix in 2024 can be questionable.

In comparison to proposed Maturity Matrix by Inge Van de Weerd, the Software Product Management Guide that is introduced in the next chapter, propose software product management maturity validation by reflecting on the problems that occur in the organisation, rather than by rating software product management functions and capabilities.

In 2013, research on software product management was conducted in Lappeenranta-Lahti University of Technology. Its main research question was: which common roles do software product managers fulfil in organizations? The result of the research is a framework that reveals the role of product managers and shows how this role can evolve by extending the level of responsibilities. The framework was developed empirically based on interviews with companies' representatives and by researching the documentation. Four stereotypical roles were identified in the studied organizations: experts, strategists, leaders and problem solvers [14].

An expert is a person who has deep expertise in some specialization but does not have the responsibility for product management activities. It is a kind of informal product manager, who works close to the software product and knows it very well. He is supporting software product development, deciding on priorities - what to build next.

The strategist is a product manager who has an impact on the company's product vision and roadmap. As the authority of the strategist grows, he becomes the third profile, the leader, who has access to product resources and can manage it.

The fourth profile is the problem solver who manages stakeholders and resolves product-related issues, but it is the management who defines the strategy and roadmaps.

Besides the characteristics of a software product managers identified in this research, authors introduce also 4 super categories of PM with properties, based on the grounded theory analysis: access to resources (ownership of the product budget, possibility to hire people, information resources); influence on the product (the orchestration of development, the definition of tactical actions, participation in strategy planning, creation of roadmaps); authority (product leadership, power of decision making); influence on collaboration (the ability to resolve problems between departments, level of communication).

Each of these categories represents the essential characteristics of a software product manager role, which define his or her areas of responsibilities and boundaries of the role. Based on these findings the Software Product Management Roles Framework (SPMRF) is developed [14]. The SPMRF serves as a tool for profiling product managers, enabling both product managers and top-level management to comprehend the responsibilities undertaken by product managers. This aids in preventing scenarios where there is a disparity in the understanding and expectations between top management and the product manager regarding the roles of the software product manager.

In 2014, Christof Ebert and Sjaak Brinkkemper defined software product management key success factors, their effect on business and the challenges that stand ahead of the product management. They also described 4 best practices for systematic product management. They stated that during the research the majority of product managers that have been found in software companies had: "a strong technical background and rather weak finance, marketing and general management skills" [15].

"Software Product Management" book, published in 2017 by H. Bernand Kittlaus and Samuel A. Fricker [2] is a review of software product management and many terms related to it like: product strategy, product planning, strategic management, orchestration of the organization's

functional areas. It also introduces the Software Product Management Framework, which is an integration and consolidation of three other frameworks that were invented in the past. The first framework that was integrated is Reference Framework for Software Product Management developed by Inge van de Weerd, Willem Bekkers, Sjaak Brinkkemper, and colleagues at the University of Utrecht, Netherlands in 2006 [28]. It describes the core activities of a software product manager in specific areas: portfolio management, product planning, software release planning, and requirements management. The second integrated framework was Pragmatic Marketing developed by Kittlaus and Clough that describes the aspects of product management, product marketing and defines the role of a product manager and product marketing manager ultimately [29]. The third component integrated was an Ebert proposal on how to manage software along the product lifecycle phases [3]. The consolidated framework is compatible with Software Product Management Guide that is presented in the next section - it focuses more on what software product management is and what are the responsibilities, when the SPMG is a framework to evaluate software product management based on existing problems in the organization.

Another study from 2018 reports that a software product manager's role and its responsibilities depend on the company size [30]. The key differences are related to the staffing of this role, its scope of responsibility, tools and techniques used as well as the mode of work with the target customers. Bigger companies might allocate some of the software product management activities to other roles that would support the product manager, whereas in smaller companies the product manager handles many SPM activities alone. The archetype of the software product manager focuses on the common elements of this role independent of the company size [30]. It was also found that a minority of product managers are responsible for the success of the product end-to-end, so they focus on core software product management activities rather than on supporting activities.

The most recent study (2023) from Nishant A. Parikh [31] synthesizes software PM responsibilities and accountability and presents it in diverse organizational contexts, with a focus on the interplay between strategic PM, technical PM, and other cross-functional roles. Strategic PM is responsible for Market analysis, Customer insights, Legal and compliance management, Roadmapping, Operations Readiness, Product Launch, Continuous market analysis and End-of-life plan. Product Definition, Product requirements engineering, Product Verification and Performance are shared responsibilities. Technical PM is responsible for Identifying Solutions, Release planning, Detailed Requirement Engineering, Accepting user stories in agile. The results indicate that strategic PM is accountable for all activities for which technical PM is responsible for. The study also explores differences in the role of Product Managers (PMs) influenced by contextual factors like company size, product development methodology, and product segment. It delves into the roles of stakeholders in product management and identifies 19 key activities for PMs, some of which are influenced by contextual factors. The findings indicate that the software product manager role is impacted by the development methodology. While the literature review did not offer clear evidence on the roles and responsibilities of technical PMs, this study reveals that a technical PM shares similar responsibilities with a Product Owner in Agile methodology.

### **2.2.1. Product Management Frameworks**

#### Software Product Management Framework [2]

Software Product Management - The ISPMA-Compliant Study Guide and Handbook is a book based on the results of the International Software Product Management Association (ISPMA) which is led by a group of experts from industry and research with the goal to foster software product management excellence across industries.

Software Product Management Framework proposed in this book indicates the main functional areas of a software organization: strategic management, product strategy, product planning, development, marketing, sales and distribution, service and support. Each area has related activities assigned.

The “core software product management” activities that the software product manager is responsible for most often, cover product strategy and product planning. Product strategy includes positioning and product definition, delivery model and service strategy, sourcing, business case and costing, pricing, ecosystem management, legal and intellectual property rights management, performance and risk management. Product planning which is also core area of software PM responsibilities includes customer insights, product life-cycle management, roadmapping, software release planning, and product requirements engineering. Strategic management also requires the participation of software product managers in corporate strategy, portfolio management, innovation management, resource management, market analysis, and product analysis.

Software product managers also orchestrate the activities related to development (i.e. product architecture, development execution, user experience design, quality management, detailed requirements engineering), marketing (i.e. product launches, value communication), sales (i.e. operational sales, customer relationship management) and delivery services and support (i.e.e operations, technical support). They support and impact by their activities those other areas making sure they perform in line with the product strategy and plan.

#### The Pragmatic Framework [29]

The Pragmatic Framework (formerly Pragmatic Marketing Framework) defines 7 areas and 37 activities related to marketing and managing technology products (Pragmatic Institute 2020). Although it does not use the term “product management” explicitly, its content covers a lot of aspects of software product management. The key aspects covered are market analysis, product strategy, business model, sales and product planning, and sales and support. However, due to the distinctiveness of software as a product, software product management includes additional aspects of user experience research, employing rapidly changing technology, iterative agile-inspired product development, and constant experimentation. This makes the Pragmatic Framework a valuable reference framework for other roles involved in product management such as the company's top management, sales and marketing, and support, rather than the software product managers themselves.

#### Product Yield Potential Radar [32]

In 2019, Timo Wagenblatt introduced Product Yield Potential Radar, PYPR, which is a detection system that determines and visualizes the yield potential and constraining factors of product success. The framework shows how to assess and visualize all dimensions that matter



to the product's success. It explains how to leverage and adapt the software product management with regard to aspects such as product viability, product development, product marketing, and software demonstrations and training, as well as more general aspects such as markets, customers and organizational maturity. PYPR provides guidance on how to improve product management performance and introduces a proven approach to holistic software product management adopted by market-defining and leading product teams (e.g. SAP).

### **2.2.2. Problems related to software product management [13]**

In 2012, Maglyas et al. provided a list of the 5 main problems related to software product management [13]. In the research, they studied 13 organizations to gain an understanding of how software product management practices were adopted. Interviews indicated the problems were not company-specific, so root-cause analysis was subsequently conducted, using the current reality tree (CRT) technique from the theory of constraints.

They identified the following software product management problems:

- Problem 1. Long Release Cycle – working in isolated units has a negative impact on time-to-market, requires more time to synchronize teams, results in many changes during the project and makes the development process unpredictable.
- Problem 2: No metrics for evaluating work – no key performance indicators (KPIs) are assigned to the product managers.
- Problem 3: Collaboration between the organization and customers – organizations are not customer-oriented; insufficient user research, user testing, and product discovery activities are performed.
- Problem 4: Short-term thinking – product teams and managers do not know the long-term strategy, the strategy is changing very often, and organizations focus on short-term actions.
- Problem 5: Trying to change instantly – it is hard to introduce all software product management activities at once but companies introduce radical changes in their organization structure and try to redesign the whole software product management activities at once.

These problems can be also analysed together with recommended guidelines in Table 2.2.

Although many publications mention some problems, the contribution from Maglyas et al. remains the only systematic attempt since 2012 to identify and describe specific software product management problems. Given the range of the software product manager's activities and responsibilities defined in the software product management frameworks, it can be assumed that problems may relate to many more aspects of product development and its lifecycle. Additionally, the majority of product managers are responsible for managing existing products that are in service and operation, instead of launching new products and looking for innovations that are related to product evolution and company growth [15]. Thus, it is expected that most of the problems related to software product management may be related to the evolution of existing products rather than to the introduction of new products to the market, however, some of them may apply to the whole product lifecycle.

A more detailed examination of the challenges in software product management is presented in Chapter 4. Specifically, within the section dedicated to studium 1, which provides an extensive exploration of the research conducted to identify pivotal issues within the realm of software product management.

### 2.2.3. Strategies to software product management [20] [3] [13]

There are also several studies on product management strategies and solutions that have been analyzed as part of this thesis.

#### *The impacts of software product management*

Ebert's study from 2007 showed that time to market, schedule adherence and handover quality improve with the strengthening of a coherent product management role [3]. Explanatory factors indicated as having a positive impact on product management have been explained and coined into guidelines towards successful product management presented in the table 2.1

**Table 2.1:** Software product management problems and solutions by Ebert [3]

ID	Guidelines towards successful product management	Key Takeways [20]
1	Business objectives and accountability	Product managers must set objectives and work to achieve them. Objectives must be measurable and connected with business needs / strategic goals. Organise a multidisciplinary product team led by a product manager (with product management, sales, marketing, and engineering competencies) so they plan, agree on the strategy and commit as a team.
2	Mastering requirements	Make sure different perspectives are considered while reviewing the requirement (description, impact, priority). Manage requirements and business objectives (plan, prioritise, agree, monitor) to assure focus Analyse the impact of each requirement to check if they support the business case. Document requirements in a structured and disciplined way. Describe both technical and business perspectives. Create Product Roadmaps. Committed roadmaps and requirements must be accessible online together with other relevant product and project information.
3	Managing risks and uncertainty	Manage risks on the level of the product line and roadmap. Use available techniques to deal with requirements uncertainty and master project risks. Prioritise requirements and plan incremental stabilization, measured by the earned value achieved.
4	Leadership and teamwork	Execute strategy consistently. Build an empowered, multifunctional core team fully accountable for the success of the product, having the mandate to "own" the project. Lead team and take short-term and long-term responsibility.

Given the range of the software product manager's activities and responsibilities defined in the software product management frameworks [2] it may be assumed that problems and solutions may relate to many more aspects of product development and lifecycle than this research pointed out.

#### *Lean solutions to software product management problems*

In 2012, Maglyas published a guide on how to apply lean practices to solve software product management problems. In the research, they studied 13 organizations to get an understanding of how software product management practices were adopted [13].

Lean management philosophy focuses on increasing value by eliminating waste. The five principles of lean philosophy are: value (providing value to the customer), value stream (mapping the value stream in order to identify and remove any steps that don't create value), flow (smoothing out the value creation process, pull (trying to meet customer needs as soon as possible), perfection (analyzing the results and planning for any implementation).

The contribution from Maglyas et al. remains the only systematic attempt to identify and describe the specific software product management problems along with solutions.

**Table 2.2:** Software product management problems and solutions by Maglyas [13]

ID	Problem	Solution	Key Takeaways
1	Problem 1. Long Release Cycle – working in isolated units has a bad impact on time-to-market, requires more time to synchronize teams, results in many changes during the project and makes the development process unpredictable.	Using Flow to Decrease Time to Market	Plan short iterations
2	Problem 2: No metrics for evaluating work – no key performance indicators (KPIs) are assigned to the product managers.	Using Value to Identify Key Performance Indicators	Set up KPIs which represent company goals for the product KPIs should be measurable Perform as a team to achieve goals Plan goals wisely
3	Problem 3: Collaboration between organization and Customers – organizations are not customer-oriented; insufficient user research, user testing, and product discovery activities are performed.	Using Pull to Develop Products Faster and with Fewer Resources	Invest time in the product analysis Start working with customers from the very beginning Work closely with customers
4	Problem 4: Short-term thinking – product teams and managers do not know the long-term strategy, strategy changes very often, and organizations focus on short-term actions.	Using Perfection to Adopt Long-Term Thinking	Develop your long-term strategy to achieve your own excellence in the area
5	Problem 5: Trying to change instantly – it is hard to introduce all software product management activities at once but companies introduce radical changes in their organizational structure and try to redesign the whole software product management at once.	Using Perfection for Incremental Changes	Introduce simple, incremental changes in the organisation

### Managing software products in a global context

In 2018 Christof Ebert published another article [33] that provides concrete practices to fertilize and evolve software product management in global teams. 75 recipients participated in the survey on global software product management. Additionally, the authors interviewed twenty representative companies. Results mention three product management-related problems, however, no solutions are proposed for software product managers to eliminate them:

- Insufficient global orchestration with unclear responsibilities and silo work which results in continuously changing focus and schedules.
- Lack of strategy and unclear strategy and roadmaps with unclear dependencies and vague feature collections, but not mapped to value creation and business cases.
- Lack of software product management discipline, which product managers evolving from technical roles and being thrown into this new responsibility without a clear competence evolution program.

Other findings from the research:

- Only 30% of interviewed companies implement the role of product manager as responsible for end-to-end success of the product.
- Over 80% of Product Managers focus on managing existing products, not new products and innovations.

- One-third of companies have Profit and Loss responsibility delegated to the product managers.

### **2.3. *Research gaps***

Frameworks and articles that were found in the literature, mainly explore the characteristics of software product management and the responsibilities of a software product manager [1][2][3] [34] [16]. Some problems related to software product management and potential strategies for these problems were proposed, but not sufficiently reviewed and quantified [13].

It was not analysed yet if and how practices that software product manager use can affect other roles and teams, especially representatives of software development teams. The literature does not currently provide frameworks, methods or tooling for improving software engineering practices in software development teams with the usage of software product management practices. The companies and teams that work without experienced software product managers cannot easily diagnose what are their key challenges and where they should focus first.

Therefore the focus of research presented in this thesis is to analyse existing problems related to software product management and verify how experienced Software Product Managers solve these problems. In this research it is validated if recognised problems, guidelines, solutions and techniques combined as a Software Product Management Guide framework can serve software development teams and influence positively their software engineering practices.

### 3. THE PROPOSED SOFTWARE PRODUCT MANAGEMENT GUIDE

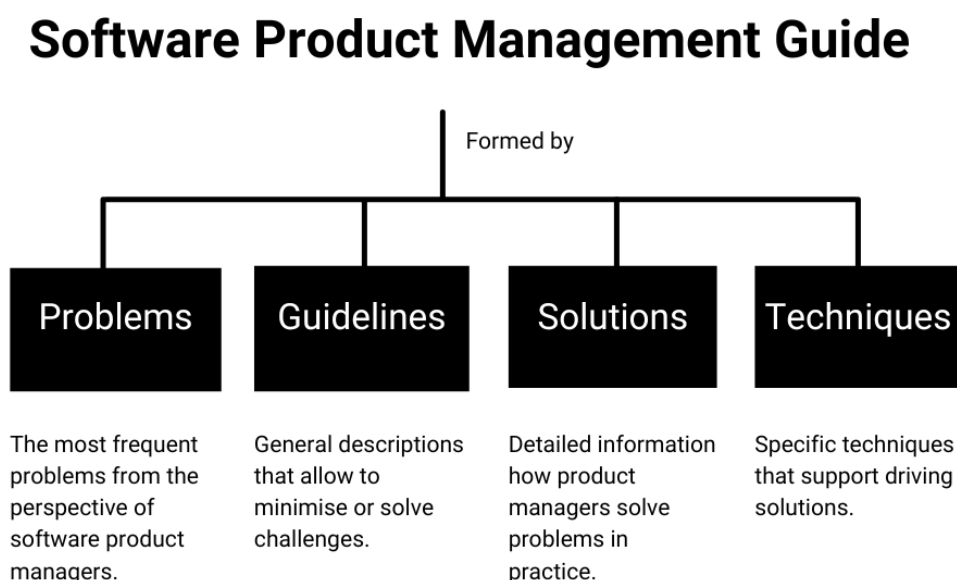
After reviewing the literature and identifying research gaps, a series of studies was planned to identify the main problems related to software product management and potential solutions to these problems. 5 different research techniques were used throughout the study: systematic literature review, interviews, surveys, focus groups, case studies. 168 software product managers participated in the whole research cycle. The studies that led to formulation of the proposed guide are described in details in chapters 4 to 6.

The results from the research allowed for the creation of a Software Product Management Guide - a framework that presents how experienced software product managers solve the problems related to software product management and lead teams to diagnose their challenges and propose a solution that can be applied in their environment. The purpose of the framework is to help product development teams to optimise their software engineering practices.

It is worth noting that the framework was prepared using the knowledge and data from software product managers working with digital products, so it does not apply to other products.

#### 3.1. *Components of the proposed framework*

The proposed Software Product Management Guide consist of the following components: problems, guidelines, solutions, techniques.



**Figure 3.1:** Software Product Management Guide

The first component of the framework is a list of **27 problems** related to software product management with perceived frequency and severity. These are the most frequent problems from the perspective of experienced software product managers.

Examples of problems:

- P74. Determining the true value of the product that the customer needs
- P35. Strategy and priorities are changing frequently
- P9. Technical debt

The second component is **Guidelines** - identified for the top 5 frequent problems. Those are high-level, general tips on what teams should focus on. Guidelines are based on discussions during focus groups where experienced software product managers participated and shared their solutions for particular problems. The number of guidelines for problems varies - there are from 3 to 5 guidelines for each problem.

Below example of Guidelines are presented for Problem P74 (Determining the true value of the product that the customer needs):

- Investigate the user's needs in-depth, use both qualitative and quantitative research.
- Collect feedback from users about the product.
- Validate ideas by working on prototypes, close to the customers.

The third component is **Solutions** - identified for the most 5 frequent problems. The solution is detailed instruction that presents how product managers solve problems in practice. Some steps in solutions can be optional or used in a different order.

The solutions that are part of the framework go through validation across software product managers for effectiveness and comprehensibility. A separate questionnaire was prepared for each of the selected problems. For each solution, respondents were asked to rate in a Likert 5-point response format (strongly disagree, disagree, neutral, agree, strongly agree) the accuracy of the following two statements: This solution is comprehensible; This solution solves the problem.

Comprehensibility of the problem solution was defined as the total percentage of "agree" and "strongly agree" responses to the first question, and perceived efficiency as the analogous percentage of responses to the second question.

Below example is presented - Solution for Problem P74 (Determining the true value of the product that the customer needs):

#### *Solution 1: Customer feedback loop*

Comprehensibility: 92,59%, Effectiveness: 66,67%

1. Integrate communication channels with the client – gather all customer feedback in one tool, e.g. Product Board (chats with clients, Intercom, reports from sales meetings). Open Slack channel where the Customer Service Team can submit issues/feedback and everyone can share their product ideas.
2. Automate as much as possible. The option to report feedback should be available in the tools other employees use on daily basis. These channels should be integrated with the feedback management tool.
3. Organise weekly review of new tickets that appeared in the feedback management tool.

4. Classify reported ideas – add appropriate tagging or connect with another ticket that already exists.
5. Each ticket, when tagged, mark with priority and complexity estimation.
6. Implement a tool in a way that it builds a hierarchy of the most frequently reported problems and needs.

The fourth component of the framework is **Techniques** - identified for the most frequent problems. During focus groups, software product managers were sharing their solutions. As part of these solutions, they mentioned supporting techniques that they use to solve particular problems. Those techniques were caught and gathered as a list of techniques to consider while working on a particular problem.

Example of Techniques for Problem P74 (Determining the true value of the product that the customer needs) are presented below.

12 supported techniques for this problem:

Interviews, Competition Analysis, Customer Feedback Loop, Product Analytics, Jobs to be done, Net promoter score, Industry Analysis, Analysis of contact points with customers, Kano Model, Value Proposition Canvas, Minimum Value Product, Personas.

### **3.2. Who can use the framework and what are the benefits?**

The framework can be used by:

- Scrum product owners or junior product managers that start their experience with software product management
- Current software product managers that want to improve their work
- Product development teams working without software product manager support, that wants to understand and improve their software engineering practices
- Product or management leaders that want to diagnose the overall state of software product management in the companies and look for strategies for potential improvements

Software Product Management Guide framework provides many benefits for these roles. First of all, it deepens the knowledge about software product management. It helps to diagnose what is the overall state of the software product management in the company. It also helps to diagnose which problems affects product development teams the most. The framework support teams in prioritisation of problems and aspects of product management they should focus on. Last but not least, it helps to define strategies for teams on how to systematize the approach to software product management and influence software product management in the company.

### **3.3. Framework application as a diagnostic tool**

The Software Product Management Guide can be used to identify key problems and help team to prioritise implementation of required strategies to solve them.

The usage of Software Product Management Guide Framework can be customised for each team, however, based on findings from the research, few best practices have been identified. In

order to use the framework, it is proposed to set up a session with the team and run through 5 steps.

List of recommended steps by the framework:

- Step 1: Analyse context. Tip: Use a General survey (Appendix 1) or ask the team basic questions about the company, their roles and their experience with software product management. This part is for the session moderator to get familiar with participants, the products they develop, and their current experience with software product management.
- Step 2: Diagnose product management state - identify with team key problems. Tip: In this exercise team scores which problems occur in their organisation and if this problem has an impact on their work (Appendix 2).
- Step 3: Select strategies - guidelines, solutions and techniques. Tip: In this phase, participants should get familiar with guidelines, solutions and techniques for a given problem, and decide which they could apply in their work. Tip: For this exercise, it is recommended to get one by one through each survey (Appendix 3) section and make a stop between them. Before going to the next section team should share their thoughts - i.e. about what interesting they found out that would be worth to apply in their project).
- Step 4: Summarise session - repeat key challenges and opportunities that were identified during the session - summarise with the team what are the top problems they should start working on and what strategies the framework recommends to try out at first.
- Step 5: Prepare an Audit Report - create a software product management audit based on learnings from the session and shared with the team or with the company leadership.

Framework guides also to invite at least 3 participants from the team, representing different roles - that would bring different perspectives and views into discussions. It is recommended to secure a minimum of 2 hours for organising the first session - this time will allow to cover a problem validation exercise and two exercises for guidelines, solutions and techniques. It is important to identify a moderator for the sessions - search for someone who has experience in software product management. Such a person will be able to organise discussions with the team, ask in-depth questions to find out root causes and challenge their thoughts. For bigger organisations, a set of sessions with teams from different departments can be considered, to understand the patterns across the organisation.

### **3.4. Framework description - problems component**

Framework defines **a software product management problem** as a concern related to some topic that affects the work of a software product manager, reduces software product management effectiveness and makes it more difficult to achieve goals. As the established software product management framework indicate [2], software product management covers many processes and responsibilities and connects to many activities carried out in the company. This is why the problems from the perspective of the software product managers may be related to a wide range of processes, activities and responsibilities of other teams/departments/roles.

There are many problems that may have an impact on a software product manager's job, and they can be related to core software product management activities or lie beyond them. The role of the software product manager and the responsibilities attached to this role differ between companies. Problems that affect this role may lay in the direct responsibilities for one product manager, and for another one, they may be external problems that are someone else's



responsibility. There are many problems that have a strong impact on the software product managers job (e.g. technical debt, which slows down the product development process and increases time-to-market), but can not be fixed by this role alone.

My studies on software product management recognised many problems of diverse impact. This is why as part of the research, not only they were defined, but also analysed how often these problems occur and what is their severity. Identification of the most frequent and significant problems allowed us to focus the research on the most important piece of work.

Table 3.1 presents identified within the Studies reported in this thesis, the 27 problems that software product managers face. The list is enhanced with the perceived frequency and perceived severity, as well as descriptions of these problems. That list of problems with the analysis of the frequency and severity was identified in a multi-step study that is described in details in chapter 4 and shared in open-access publication [19].

**Table 3.1:** Evaluation of the perceived frequency and perceived severity of the selected software product management problems [19]

Problem ID	Problem name	More details about the problem	Frequency	Severity
P74	Determining the true value of the product that the customer needs	It is challenging to run extensive research and work iteratively with teams to understand the customer needs, scaling opportunities and customer willingness to pay for the product.	72,7%	69,3%
P35	Strategy and priorities are changing frequently	When the strategy is changing frequently, product managers and teams struggle with prioritisation and can't achieve outcomes, as direction is changing too often.	71,6%	70,5%
P9	Technical debt	Challenge with developing and maintaining the existing code that is slowing down product development and makes it even more difficult to prioritise the Product Roadmap.	70,5%	64,8%
P64	Working in silos, problems with communication, synchronization between teams	Initiatives run across different departments require more discipline to align teams around common goals, synchronise them and organise communication that will support team rather than slowing them down.	69,3%	65,9%
P69	Balancing between reactive and proactive work.	For mature products, it is always a challenge to prioritise research, innovation and new value for customers against a queue of bugs, escalation, and maintenance-related work.	69,3%	63,6%
P26	Lack of support for any research (no resources allocated to the team)	No resources allocated to the team that would support teams in gaining more knowledge to decide on priorities through research	65,9%	63,6%
P85	Lack of automated testing	No automated testing, resulting in lowered quality and impact on customers who are using the product	65,9%	62,5%
P82	Product manager role not clearly defined and communicated in the organization	Other teams and individuals don't understand what the product manager role is about, what the responsibilities, objectives, decision-making level	63,6%	58%
P48	Lack of user research	Teams don't spend enough time on research which results in no proper understanding on the problem space and customer side	61,4%	64,8%
Continued on next page				

Problem ID	Problem name	More details about the problem	Frequency	Severity
P51	Roadmap focused on features instead of goals and business value	Product Roadmap represents the features that will be developed instead of the problems that will be solved or objectives that are going to be met	61,4%	59,1%
P18	Lack of market research, no understanding of business and trends in the industry	Teams don't have knowledge about what is happening on the market, what are the trends to be able to search for opportunities and valid problems to be solved	55,7%	55,7%
P8	Different expectations of individual stakeholders regarding product management communication (high/low level)	Hard to meet all expectations from stakeholders from different teams and with different seniority levels	55,7%	52,3%
P6	Lack of continuous integration and delivery (impact on 'time to market')	No proper Engineering standard in place that results in more time needed to release the product to customer.	55,7%	48,9%
P25	Limited access to users in order to do research	Not easy to find potential customers or existing users to perform a research, get new knowledge or validate solutions	54,5%	59,1%
P16	Lack of synchronization between product management and marketing and sales units	Product, marketing and sales work separately, don't share same goals and are not aligned.	54,5%	52,3%
P1	No company strategy	When there is no strategy, product managers and teams struggle with prioritisation, don't see the long term picture and can't achieve outcomes. Requirements and expectations from stakeholders change very often.	53,4%	61,4%
P44	Teams are not autonomous and self-organized	When teams don't have enough autonomy, it is difficult to organize work, responsibilities. Decision-making is dispersed.	53,4%	58%
P10	Lack of user analytics data	Teams don't track behavior data from web and mobile applications to achieve a holistic view how user interacts with the product and what is their user experience.	48,9%	54,5%
P19	Teams are not Agile, they just follow rules and do not use experimentation and a learning process	Teams work in iterations and follow some rituals, but as they proceed with work, they don't adapt the process and product. Examples: They don't increase their knowledge about customer problems/needs, don't value and prioritise decisions that allow them to learn further.	47,7%	51,1%
P13	The team's lack of motivation due to a lack of understanding of why they are doing things – no commitment to achieve goals	Teams don't understand why they are running some initiatives - they don't believe in the purpose	47,7%	46,6%
P41	High expectations from external partners, which are not possible to be met	Teams struggle with requirements from external partners	47,7%	45,5%
P3	Wrong data analytics setup	Gaps in the user analytics data that make it difficult/impossible to analyse and learn from it	44,3%	44,3%
Continued on next page				

Problem ID	Problem name	More details about the problem	Frequency	Severity
P12	Low software quality	Quality challenges impact customers who are using the product	43,2%	45,5%
P78	Lack of skills to use and analyze the data	Teams don't have enough knowledge on how to work with user analytics data	43,2%	43,2%
P43	Lack of trust in the product team (micro-management)	Teams don't have enough trust to decide on priorities and solutions that they are going to invest in	42%	51,1%
P27	Unqualified team members (individuals)	Teams struggle with competences of individual team members	42%	50%
P56	Price management is always experimentation burdened with risk	Teams struggle with decisions about product pricing plans	36,4%	29,5%

### **3.5. Framework description - Solutions, Guidelines and Techniques components**

In the following sections, recognized guidelines, solutions and supporting techniques for the most frequent software product management problems are presented. Each subsection presents results for one problem, according to the following scheme:

- Short summary of recognised strategies – general thoughts and summary about the guidelines and identified solutions from the framework author.
- Guidelines – key insights on how to tackle specific problem, based on solutions and discussions during focus groups.
- List of identified solutions to the problem, where each solution is presented in a subsection, along with a measure of average comprehensibility and effectiveness, followed by a detailed description.
- List of identified supporting techniques.

The collection of solutions (with the analysis of the comprehensibility and effectiveness), guidelines and supporting techniques were identified in a multi-step study that is described in details in Chapter 4 and shared in open-access publication [20].

#### **3.5.1. Problem: Determining the true value of the product that the customer needs (P74)**

The majority of solutions recommended for this problem underlined the need of working closely with customers to find out their's needs. Both quantity, quality and mixed methods were recommended to use in the research. It is also important to validate the ideas as soon as possible with customers and gather feedback constantly.

Guidelines for this problem:

1. Investigate the user's needs in-depth, use both qualitative and quantitative research.
2. Collect feedback from users about the product.
3. Validate ideas by working on prototypes, close to the customers.

To solve this problem, framework proposes the following 7 solutions.

### Solution 1: Customer feedback loop

Comprehensibility: 92,59%, Effectiveness: 66,67%

1. Integrate communication channels with the client – gather all customer feedback in one tool, e.g. Product Board (chats with clients, Intercom, reports from sales meetings). Open Slack channel where the Customer Service Team can submit issues/feedback and everyone can share their product ideas.
2. Automate as much as possible. The option to report feedback should be available in the tools other employees use on daily basis. These channels should be integrated with the feedback management tool.
3. Organise weekly review of new tickets that appeared in the feedback management tool.
4. Classify reported ideas – add appropriate tagging or connect with another ticket that already exists.
5. Each ticket, when tagged, mark with priority and complexity estimation.
6. Implement a tool in a way that it builds a hierarchy of the most frequently reported problems and needs.

### Solution 2: User testing sessions – testing prototypes

Comprehensibility: 96,30%, Effectiveness: 88,89%

1. Form a hypothesis and create a business case.
2. Prepare a workflow for the process with all dev, UX.
3. Prepare prototype/wireframes for testing.
4. Organise user testing session with the target group – i.e. a task for test participants to go through a new registration process. Record this session.
5. Research results, identification of pain points, understanding of client's needs.
6. Recommend further steps for the top-level management.

### Solution 3: In-depth interviews with potential/future users of the product

Comprehensibility: 100%, Effectiveness: 85,19%

1. Before starting any work, it is the end-users who define problems and needs – collect information and requirements from them.
2. Verify information about customers with stakeholders inside the company (what we must, can, should, cannot provide as part of this application).
3. Review of competitors and their products/systems.
4. Verify assumptions with end users (i.e. by showing the prototype/wireframes).
5. Prepare of a list of functionalities satisfying the needs of customers and preparation of the MVP scope (selection of those elements that must be included in it).

#### Solution 4: Data analysis and in-depth interviews with clients.

Comprehensibility: 92,59%, Effectiveness: 88,89%

1. Analyze user metrics, or define such metrics if they do not exist. Measuring the frequency of using functionalities on the basis of events, tools monitoring the front of the application, such as Heap, etc.
2. It is necessary to analyse the topic further through face-to-face interviews with customers – frequent clicks do not always mean frequent use of the feature, but may mean that something is unclear or that it is loading poorly. Therefore, you need to combine quantitative and qualitative analysis. Conversations with customers in order to understand their needs and how the product addresses these needs and where it does not.
3. Instead of an in-depth interview, an even better solution is to observe the user: the user gets tasks to be performed and we record his paths.
4. Define “personas” and “jobs to be done” – defining the product’s “golden flows”.
5. Using the collected data before planning the Roadmap (data as starting point to make ideas real). Always refer to the data.
6. Continue hypothesis validation and use iterative “feedback loop”.

#### Solution 5: Validation of whether we are solving the customer's problem based on simple Proof of Concept

Comprehensibility: 92,59%, Effectiveness: 88,89%

1. Prepare a simple version of the solution that solves user problems – fixed values, no automation, manual operation, use of available frameworks to minimize the work to be done.
2. Check if the solution has a chance to work and help users at the same time.
3. Organise in-depth interview sessions in order to create a list of optimization to do and improvement in subsequent versions of the solution.

#### Solution 6: Jobs to be done

Comprehensibility: 85,19%, Effectiveness: 88,89%

1. Make a thorough analysis of the problem that we want to solve.
2. Investigate how the target group is solving a given problem now.
3. Investigate whether a given problem is really important and who has the greatest motivation to solve it.
4. Analyse the strengths and weaknesses of the solutions that the target group is currently using when trying to solve a given problem.
5. Define the proper positioning of our product.
6. Determine “Jobs to be done” and validate hypotheses.

### Solution 7: Value proposition canvas

Comprehensibility: 85,19%, Effectiveness: 81,48%

1. Create a customer profile – map jobs to be done, pains and benefits sought.
2. Create a description of the solution.
3. Make Value Proposition Canvas as a living artefact – it changes like the needs and actions of our clients.
4. Use only validated information (through research, observation, and feedback analysis).

### 12 supported techniques for this problem:

Interviews, Competition Analysis, Customer Feedback Loop, Product Analytics, Jobs to be done, Net promoter score, Industry Analysis, Analysis of contact points with customers, Kano Model, Value Proposition Canvas, Minimum Value Product, Personas.

### **3.5.2. Problem: Strategy and priorities are changing frequently (P35)**

The solution for this problem recommends working with stakeholders to discuss the strategy and main challenges in order to set up clear business goals. Product managers also suggest showing the uncertainty of the future by creating visual roadmaps without strict timelines. Last but not least it was mentioned a few times during focus groups, that if priorities change frequently it is a strong signal that there might not be a strategy at all or the employers are simply not informed about the reasons behind priority changes coming from the top, and how those decisions relate to strategy.

Guidelines for this problem:

1. Create product visions with the most important stakeholders, define the main challenges and business goals.
2. Reflect the uncertainty of your strategy in the form of a visual roadmap.
3. Develop a strategy with key stakeholders – use results from the research as inputs to the strategy.
4. Make sure your strategy is communicated.

To solve this problem, framework proposes the following 6 solutions.

### Solution 1: Strategy and product-centred process

Understability: 100%, Effectiveness: 96,55%

1. Cooperate with the stakeholders on the product vision.
2. Together with the stakeholders elaborate the main business goal/ business challenge to achieve on the market to achieve the state described in the product vision.
3. Together with the team divide the goal/challenge into smaller measurable goals to achieve in the product to get closer to the main goal.

4. Clearly describe the current state of the product.
5. Aim to employ and build a bigger product team.
6. Develop the product-centred process starting from problem collection up to solution development including prioritization, transparency and roadmap.
7. Promote the process and strategy.

#### *Solution 2: Yield potential (incremental) / cost of delay + urgency profile*

Understability: 86,21%, Effectiveness: 65,52%

The solution is to analyse the impact and define (estimated) the value and urgency of new features. It requires a substantial initial effort to estimate many features at once e.g. while making a quarterly plan. Then we should require the same exercise for every new idea. A trade-off analysis of every new idea and the current backlog should be performed, and the capacity increase should be discussed. To implement this solution:

1. Define the expected potential for every feature (preferably yield potential).
2. Define the urgency/importance of every new feature/project (will introducing the feature in 6 months make sense? Will it change the yield potential?) – urgency profile.
3. Show the trade-offs to the management and other teams e.g. we are developing A, B, C – if you want to develop D, we have to drop another feature.

#### *Solution 3: Add to the roadmaps items like custom-tailored deployments or integrations*

Comprehensibility: 82,76%, Effectiveness: 62,02%

This solution addresses the problem when the onboarding of new big customers was not included in the yearly plans and finally, it won over other highly prioritised tasks. With B2B products, we often face Enterprise customers that use custom-tailored solutions, thus the increased onboarding effort might win over other tasks.

1. Add planned custom-tailored deployments and integrations with customers' systems (primarily non-standard ones) to the yearly/quarterly plans. It requires good synchronization of sales and product departments.

#### *Solution 4: Now, Next, Later roadmap*

Comprehensibility: 96,55%, Effectiveness: 86,21%

Assuming the product is in the growth phase and the changes to the strategy are justified, frequent strategy changes should not be the problem – we are searching for new product solutions, new ways to win on the market. We should not fight it, we should embrace it.

1. Develop “soft roadmaps”. In case of frequent changes arrange with the management to define the roadmap in “Now, Next, Later” blocks instead of time. Show uncertainty on the roadmap. This roadmap makes the product more flexible and adaptable to the changing environment.

#### *Solution 5: Define strategy*

Comprehensibility: 89,66%, Effectiveness: 62,07%

If the strategy is changing frequently, then possibly it does not exist at all.

1. Carry out research among the employees and verify if they know the company strategy.
2. Define the strategy.
3. Communicate the strategy and define the goals consistent with the strategy.

#### 10 supported techniques for this problem:

Customer Interviews Workshop with internal stakeholders Impact Analysis, Product Analytics, Product Vision, Product Roadmap, OKRs, North Star Metrics, Strategy.

### **3.5.3. Problem: Technical debts (P9)**

The solutions described below show that even though is not the main responsibility of the product manager to decide on the technical architecture and challenges, he can still influence the teams to take care of technical debt when needed.

Guidelines for this problem:

1. Build debt awareness in the organization, show-specific measures and numbers to educate stakeholders on what is the challenge and why it is worth an investment.
2. Initiate discussions about technical debt with the team (during scrum ceremonies or additional meetings).
3. Prioritise initiatives related to decreasing technical debt in the roadmap or plan constant time each sprint to take care of it.

To solve this problem, framework proposes the following 9 solutions.

#### Solution 1: Smuggling

Comprehensibility: 85,71%, Effectiveness: 71,43%

1. Smuggle debt tasks as mini-tasks, items of other tasks, when planning a sprint.
2. Cover Technical-related topics regularly with the team and communicate during the meetings (retrospective, sprint review).
3. Meeting once a month to discuss what needs to be done in the context of technical debt. The team's task is to plan the order of individual, required implementations.

#### Solution 2: Measure and build awareness

Comprehensibility: 100%, Effectiveness: 71,43%

The biggest challenge is that the business does not understand the cost of (bad) technical debt. The solution is to measure the debt and build awareness in the organization.

1. Measure your debt. Few options:
  - Use ready-made tools, like Sonarqube that shows how much technical debt the code contains.



- Use one of the following measures, for example, division of maintenance and development tasks, the number of bugs reported, application performance. It is worth looking for good examples of measuring debt on the market and choosing a few of the most important.
  - Set up an Epic (folder for tasks) in Jira which includes all technical-debt related issues.
2. Build debt awareness in the organization, educating what we are dealing with.

#### *Solution 3: A constant topic at the Sprint Review meeting*

Comprehensibility: 95,24%, Effectiveness: 71,43%

Create a space to talk with the team about technological debt.

For example, Technical debt can be one of the points to be discussed during the Sprint Review – so that it is also a place to exchange information about technical debt between the team and the PO. During the Sprint Review: a summary of what debt “loans” we took in a given sprint, but also what part of the debt we managed to pay off.

The PO should be aware of the debt and the consequences of such and not other decisions. In turn, the team should feel that this topic is important is not hidden under the rug.

#### *Solution 4: Product Owner taking the initiative*

Comprehensibility: 100%, Effectiveness: 90,48%

1. Make an conversation with the team, showing that we are aware of what technical debt is and what consequences it has.
2. Discuss with the team in order to identify those places that are the greatest “brake” for the team, e.g. during refinement.
3. Arrange Product Backlog Items for debt repayment and prioritizing them.

#### *Solution 5: Product Roadmap – analysis and update*

Comprehensibility: 80,95%, Effectiveness: 80,95%

1. Analyse technical debt with the team and define areas to be addressed.
2. Estimate work required to reduce technical debt in specific areas.
3. Establish priorities to be addressed taking into account the estimate and impact on the product (technical and business).
4. In the case of larger topics, add initiatives related to the reduction of technical debt to the Product Roadmap. While working on a given area in connection with another ongoing initiative, make an overhead to also deal with the technical debt.

#### *Solution 6: Managing technical debt*

Comprehensibility: 85,71%, Effectiveness: 85,71%

1. When a project is starting from scratch – agree with the team and technical leaders that any noticed or conscious shortcomings are reported and aggregated within, for example, one epic. With an inherited product – analyse the existing state and prepare an initial optimization list.

2. Educate customers – why it is so important to identify technical debt (impact on the product/business/users) and what value will the optimisations (i.e. refactoring, library uplift) bring ultimately. Build awareness of debt as something normal in the software development process that needs to be managed. Share examples (i.e. LinkedIn rewritten the entire core from Ruby to NodeJS because at some point development/maintenance/scaling was too cumbersome/expensive).
3. Implement the rule – leave the code better than you found it.
4. Reserve time for regular debt reduction as part of the sprint. Include the necessary refactors / corrections in the task estimates.

#### *Solution 7: Taking initiative and proposing change*

Comprehensibility: 85,71%, Effectiveness: 76,19%

If noticed there are too many problems due to technical debt, try to find out with the team how you can simplify the system, architecture, because it may happen even that a simple migration (i.e. database), decreasing process complication, or automation can enable to continue development and reduce the number of resources required to reduce technical debt.

Ask yourself a question: What can I do, as a product manager, to have an impact? Take the initiative and propose a concrete change in a specific area.

#### *Solution 8: Prioritisation + communication transparency + ownership*

Comprehensibility: 80,95%, Effectiveness: 76,19%

Context: project after the MVP phase, before the development of new functionalities, has a debt in the form of e.g. 2 months of required bug fixing due to software vendor fault, blocking the effectiveness of further implementations.

1. Communicate transparently the problem to the client.
2. Take responsibility for the current state of implementation and repair costs.
3. Prioritize functionalities that have to be fixed first.
4. Identify new functionalities that can be implemented in parallel, without duplicating debt (if possible).
5. Establish steps to prevent similar incidents in the future (bug fixing, code review, unit and E2E + manual tests specified % of the sprint time).

#### *Solution 9: Constant % of time spent on technical debt*

Comprehensibility: 95,24%, Effectiveness: 76,19%

1. Analyse the volume of technological debt (in the backlog) – i.e. external company audit in the field of system security, or internal analysis.
2. Make ongoing backlog updates with tasks related to technological debt. Additionally, once a quarter / half a year, brainstorming to define new things / analyze the size of technical debt.
3. Estimate work for every individual element of technical debt and prioritise.

4. In each sprint, allocate a certain amount of time (eg 10%) to tasks related to technological debt. Pick up technical debt tickets for each sprint (depending on severity and criticality).

#### Solution 10: The boy scout rule - clean up

Comprehensibility: 85,71%, Effectiveness: 66,67%

When implementing the change or development of a given area, allow the team to clean that area in which they are currently moving (in a predetermined manner and to a predetermined extent).

None supported techniques for this problem.

### **3.5.4. Problem: Working in silos (problem with communication, synchronisation between teams) (P64)**

Product managers can not change the company structure, but they can still minimise its impact on the teams and product management process. They recommend focusing teams around defined strategic goals and creating a synchronisation process. When teams are not working closely it is also crucial to set up communication channels to keep everyone updated.

Guidelines for this problem:

1. Define strategic product goals to which particular teams would contribute.
2. Work on transparent and periodic communication to inform about goals, updates, learnings.
3. Set up rituals in order to synchronise the teams: product demos, kick off meetings before new projects start, regular meetings, roadmapping process.

To solve this problem, framework proposes the following 11 solutions.

#### Solution 1: Product team demos

Comprehensibility: 100%, Effectiveness: 71,43%

Introduce a culture of demoing in the organization. Show the entire product at the Demo meeting (open for all in the company)

1. Product teams sign into the agenda if they want to show something.
2. Each team shows what they achieved, other participants may ask and provide feedback. Teams show the working product, not documents.
3. The Demo meeting can be introduced as a regular weekly meeting. A dedicated Slack channel can be created for the people to share information about the demo, ask questions.

The demo meeting can include current product changes as well as current topics in the discovery/research phase.

#### Solution 2: Develop clear rules for collaboration, define processes and responsibilities, promote teamwork

Comprehensibility: 90,48%, Effectiveness: 90,48%

1. Involve the team members in making key decisions e.g. on architecture.

2. Define the process that is comprehensible, transparent and easy to apply.
3. Apply a single source of truth in the context of both requirements and organizational culture.
4. Promote teamwork, even in code programming, and ensure the team sees the value in teamwork.
5. Carry out the daily meetings correctly – moderated, including status, goals and problems.
6. Coordinate between teams.

#### Solution 3: Strategic goals

Comprehensibility: 95,24%, Effectiveness: 80,95%

1. Define common and product goals to which particular teams would contribute.
2. Define goals for the teams responsible for parts of the product.
3. Organise regular meetings of the product managers.

#### Solution 4: Increase team autonomy

Comprehensibility: 95,24%, Effectiveness: 80,95%

1. As a Product Owner build the competencies, ensure sharing of knowledge. Find other teams that have the competencies needed and invite more experienced people to less experienced teams. Build an autonomous team with all the necessary competencies.
2. “Review” meeting to analyze the implemented solution from the development, customer and UX point of view.

#### Solution 5: Introduce process to synchronize the work of different product teams

Comprehensibility: 95,24%, Effectiveness: 76,19%

Introduce a process for synchronization with other teams working on the same goal/area.

1. Organise regular high-level meetings where team leaders share the knowledge and update the status/scope of their work.
2. Organise Regular (weekly or bi-weekly) meetings of a particular product area. The participants are: product managers and technical leads (engineering managers) from different product teams that work within one area. The goal of the meeting is to identify the interdependencies among the teams working on that area.
3. Introduce the elements of the SAFe framework for synchronization.

Regular meetings of the representatives of different teams decrease the barriers of ignorance and distance between the teams in the area.

#### Solution 6: Create multidisciplinary teams for initiatives

Comprehensibility: 95,24%, Effectiveness: 71,43%

Create multidisciplinary teams (e.g. PM, EM, marketing, analyst, PMM) for particular initiatives oriented on achieving a concrete goal.

1. A person having a given role e.g. PM can participate in many multidisciplinary teams at the same time which broadens the context of the work of that person.
2. The team has a goal assigned, can be created for a particular period of time or for a given initiative.
3. The created multidisciplinary team has its own Slack channel and meetings, sets the priorities of work together, carries out experiments.

In this case, a top-down activity is required to introduce this solution. A pilot multidisciplinary team can be created bottom-up and evangelized and promoted in the organization.

#### *Solution 7: Regular meetings of POs and project managers*

Comprehensibility: 90,48%, Effectiveness: 80,95%

Regular status meetings of Owners and Project Managers of particular teams to synchronize projects and common roadmap.

#### *Solution 8: Working together on the roadmap by all teams involved in product development*

Comprehensibility: 85,71%, Effectiveness: 76,19%

POs and project managers of particular teams synchronize the topics between teams when the roadmap for upcoming quarters is designed.

#### *Solution 9: Kick-off meeting before project start*

Comprehensibility: 90,48%, Effectiveness: 76,19%

1. Start large projects with a kick-off meeting to which invite all potential stakeholders of the project (even if there will be too many – good agenda should filter out unnecessary participants).
2. Define key stakeholders during the meeting i.e. those interested in the project. Take into account not only the project area but also assumed impact/collateral (e.g. when introducing functionality that will not apply to a certain market, the manager of this market should become a stakeholder) and the KPI-based project impact.

#### *Solution 10: Transparent and periodic communication*

Comprehensibility: 95,24%, Effectiveness: 66,67%

Prepare a short description of key things that happen in the product and publish them on public channels e.g. Slack, MS Teams regularly (weekly or bi-weekly).

#### *6 supported techniques for this problem:*

Kick-off meeting, Product Goals, Demo meeting, SAFE implementation, Multidisciplinary teams, Periodic communication.

### **3.5.5. Problem: Balancing between reactive and proactive work (P69)**

To make sure teams spend enough time on proactive work, Product managers recommends finding a way to guarantee the time for proactive initiatives at the level of team goals. They also

believe the time is required to run a research in order to create a product vision. In case when it is not possible to involve all teams in proactive initiatives, it is recommended to create a multidisciplinary team dedicated to running product discovery.

Guidelines for this problem:

1. Guarantee the team a minimum share of proactive work at the level of OKR or team goals.
2. Spend time to plan the future – create a vision or north star. Setting product development goals that support this vision and searching for innovations.
3. Analyse impact and prioritise initiatives or backlog items in order to define these activities that will bring the greatest value to the user (whether they come from a proactive or reactive pool).
4. Create a multidisciplinary team dedicated to running Product Discovery.

To solve this problem, framework proposes the following 8 solutions.

#### *Solution 1: Setting goals for product development - devote time to reactive and proactive work*

Comprehensibility: 94,12%, Effectiveness: 76,47%

Adopting goals for product development – dividing % of the time spent on reactive work (bugs, technological debt) and proactive (work on the set goals of OKRs, developing new things). Guarantee a minimum share of proactive work at the level of OKRs (Objective and Key Results) or team goals.

1. Define OKRs / OITs based on the product strategy.
2. Mark the elements of the Product Backlog that are connected with OKRs and those that do not relate to them.
3. Plan Sprint Backlog in a way that 70% of tasks directly contribute to strategies / OKRs, 30% for bugs or other more ad hoc ideas.
4. Measure real team effort devoted to the development of new functionalities and maintenance.

Reactivity in the mature phase of the product is hard to limit, so it is worth taking into account the need to increase resources if development/innovation work is expected.

#### *Solution 2: Forecasting*

Comprehensibility: 82,35%, Effectiveness: 64,71%

The visual building approach – building everything with the future in mind, not the present.

1. Based on the growth history of products, set a growth forecast. Use input from reports, bugs, and customer interviews.
2. Propose solutions for the state in which it is believed customers will find themselves in a year, two or more.
3. Validate that vision with selected customers.
4. Verify that vision constantly using observations from ongoing initiatives.

### Solution 3: Continuous prioritization (e.g. by RICE method)

Comprehensibility: 88,24%, Effectiveness: 82,35%

Continuous prioritization of planned initiatives or backlog items in order to define these activities that will bring the greatest value to the user (whether they come from a proactive or reactive pool).

For prioritization, use one of the methods, i.e. the RICE to evaluate items based on four variables: Reach, Impact, Confidence, Effort

### Solution 4: Introduce product discovery process in the organisation

Comprehensibility: 94,12%, Effectiveness: 82,35%

1. Distribute delivery work from discovery.
2. Create a separate Jira board, separate discovery targets and assume that 20% of PM and UX / User researchers work is related to discovery.

### Solution 5: Customer advisory board

Comprehensibility: 100%, Effectiveness: 64,71%

Constant contact with product ambassadors, having a group of customers to talk with. Using customer advisory board to test hypotheses, analyse the impact, confidence. Estimating work with the team as a next step. Having a proactive attitude.

So even rigid requirements from the business, the CEO, doing your job well, i.e. validation, value, looking for innovations. Understanding customers to have a bargaining chip in the discussion and hard arguments which show “as is” facts.

### Solution 6: North star

Comprehensibility: 100%, Effectiveness: 82,35%

1. Set product development goals / supporting innovation (Roadmap, OKR) (prioritization).
2. Keep teams/work focused on the goals defined. Synchronise work between teams.
3. Keep cooperation close with clients and define their long-term needs (input to the roadmap).
4. Cooperate with internal stakeholders.
5. Visualize the goal with mockups.
6. Define the speed of teams and define the commitment to work for current clients (dev/bugfix).
7. Monitor performance of the application and user behaviour.
8. Remove unused functionalities from the product (simplification of maintenance and further development)

### Solution 7: Define goals (OKRs) and product roadmap

Comprehensibility: 94,12%, Effectiveness: 88,24%

1. Define high-level goals (OKRs – Objective and Key Results) and going down with them to the teams. Designate OKRs for the “team”, include both discovery and delivery objectives to balance the ratio of proactive and reactive work.
2. Define the Product Roadmap.
3. Set up sprint goals in line with the goals and roadmap.
4. Plan work in accordance with the purpose of the sprint.
5. Designate a person per sprint responsible for addressing unplanned tasks (eg defects) or leave a buffer in the team's 'capacity' for addressing these tasks.
6. Build assertiveness in the development team.
7. Say 'NO' to tasks inconsistent with the goals – use appropriate argumentation.

#### *Solution 8: Dedicated team to make a product discovery*

Comprehensibility: 94,12%, Effectiveness: 70,59%

Create a dedicated, multidisciplinary team for Product Discovery (consisting of various competencies: eg ux researcher, marketing manager, product manager).

The goal of the team is to constantly plan and implement research, explore problems in order to identify opportunities.

1. Focus on analyzing and explore the problem that we solve for the client and looking for insights that can be used in development.
2. Organise a constant R&D process in which team conduct research, identify opportunities and test potential solutions. Create a list of opportunities, potential solutions, backlog of planned experiments. Management and summary of experiments (success criteria, conclusions).
3. Define what teams should build to deliver value to the user in such a way as to deliver business value.
4. Test and experiment, search for solutions 10x better rooted in the company's culture.

#### *Solution 9: Implementation of maintenance and customer service management standards*

Comprehensibility: 100%, Effectiveness: 76,47%

1. Analyse the causes of incidents.
2. Learn from incidents in order to minimize their occurrence in the future.
3. Support and optimize processes to minimize the need for reactive work.

#### *Solution 10: Define the problem, prioritise and create roadmap*

Comprehensibility: 94,12%, Effectiveness: 88,24%

The strategic team consist of product manager, designer, analyst, engineering manager) – responsible for research, benchmarking. A separate backlog is being managed, focused on Discovery related topics.

Steps to define problem:



1. Collect facts (feedback from users, internal and external stakeholders; benchmarking, tracking data, IDI with users) – cooperate with Product Designer, Product Analyst, Researchers, EM.
2. Create a list hypotheses to be tested, plan experiments.
3. Conduct experiments and confirming the problems and proposed solutions.

Steps to prioritise:

1. Collect problems with proposed solutions.
2. Organise a workshop using MOSCOW method – gather all stakeholders and a person who will be the user's advocate, prioritize in terms of what must, should, could, would be done, from the perspective of the user, business and other important perspectives.
3. Arrange prioritized solutions on a solution pyramid to answer the question of where (base, middle, tip) these solutions rank for the problems confirmed in the experiments.
4. One Roadmap – set up a strategy to deliver in the shortest possible time the greatest value for the user, business, and all others. Setting up precise goals.

#### 17 supported techniques for this problem:

Workshops, Customer Interviews, Market Analysis, Customer Feedback Loop, Design Sprint, Impact Analysis, Product Goals, Product Roadmap, RICE prioritisation, North Star Metric, Customer Advisory Board, OKRs, Product Vision, Moscow Analysis, Industry Analysis, Jobs to be done, Strategy.

### **3.6. Framework - Supporting Techniques**

Throughout the focus groups, software product managers actively shared their solutions to specific problems. As they presented or described these solutions, they also referenced supporting techniques integral to resolving particular challenges. These supporting techniques were keenly observed and documented by the moderators, forming a comprehensive list of techniques to consider when addressing specific problems.

Table 3.2 meticulously outlines all the identified techniques discussed in the focus groups, particularly focusing on the five most prevalent software product management problems. Each technique is accompanied by a concise description and an indication of the specific problem it effectively addressed. The order of listing techniques follows the frequency of problems addressed.

**Table 3.2:** List of techniques identified in the study [20]

Technique	Description	Problem addressed
Customer interviews	Interviews conducted with customers, with the intention of receiving potentially helpful feedback and insights [35]	P74, P35, P69
Competition analysis	Identifying and analysing the competitors' strategies in order to find your own advantages/weaknesses and gain a better understanding of the market and its trends [36]	P74
Continued on next page		

Technique	Description	Problem addressed
Customer feedback loop	Constantly collecting, analysing and applying customers' feedback, then reaching out to them in order to let them know how their suggestions helped improve the product, making them feel heard and improving their overall experience [37]	P74, P69
Personas	Creating fictional characters based upon your research which represent your customers in order to gain a broader perspective and understand other people's needs and expectations [38]	P74
User observation	Observing users interacting with the product to better understand its usability and the user experience [39]	P74
Product Analytics	Analysing how users interact with the product in order to optimise it and improve the overall experience [40]	P74, P35
Jobs-to-be-done	Identifying the tasks customers want the product to help them with and improving it in those areas [41]	P74, P69
Net promoter score	Gauging customer loyalty by asking the customers whether they would recommend the product to others [42]	P74
Industry analysis	Analysing the industry in order to better understand a company's position in it, its strengths, potential threats and areas most worth improving upon [43]	P74, P69
Analysis of contact points	Analysing of the points of contact between the company and the customer, in order to find any potential flaws within them and fix them, improving the customer experience [44]	P74
Kano Model	Dividing the attributes of the product into 5 different groups based on their necessity, eliminating those which hinder or don't significantly affect the user experience and improving upon those which are most valued by customers [45]	P74
Value proposition canvas	Identifying the users' jobs-to-be-done, pains (the annoyances they encounter while trying to complete their jobs-to-be-done), classifying them as light or severe, and what benefits they could reap from your product. Then identifying the services your product is going to deliver, what benefits it brings to the users and how it will help them relieve the aforementioned pains [46]	P74
MVP	The acronym stands for Minimum Viable Product. In this technique you begin by releasing a bare bones, yet functional version of your product and only then changing and improving it based on customer feedback, in order to avoid wasting time on developing unnecessary features [47]	P74
Workshop with stakeholders	Inviting the stakeholders (people affected, having a direct interest or in any way involved in the problem at hand) and the gatekeepers (the ones controlling the access to the people and resources required) to a short workshop in order to receive their input and achieve a common consensus [48]	P35
Impact analysis	Thinking through the full impact a change may have on the company and product to avoid and/or prepare for any future problems [49]	P35, P69

Continued on next page

Technique	Description	Problem addressed
RICE prioritisation	The acronym stands for Reach, Impact, Confidence and Effort. This technique entails scoring your project based on its reach (an increase in customer activity in an area of your choice in a specific time frame), impact (for example on the number of user sign-ups), your confidence in it and the effort its implementation will take [50]	P35, P69
Product vision	The idea of how the product should look in the future: its features, target audience etc. [51]	P35
Product roadmap	A plan of the product's development, which involves the tasks at hand and its vision, adhered to by all the teams involved in the process [52]	P35, P69
OKRs	The acronym stands for Objectives and Key Results. The technique entails setting an objective, the completion of which can be indubitably verified using the key results, which were supposed to be achieved [53]	P35
North Star Metrics	A metric that captures the core value you aim to provide to the customers, very helpful in staying focused on your main goals and effectively improving upon your product [54]	P35, P69
Strategy	The decisions made and actions undertaken by a business to achieve a larger goal in the long-term [55]	P35
Kick-off meeting	A meeting held before the project's execution begins with the teams involved, where a vision of the project and an initial plan for its development are finalised [56]	P64
Product goals	Key performance metrics used to measure its progression and success [57]	P64, P69
Demo meeting	Meetings held with the people involved in the project to present its latest demonstrable version in order to update them on the progress and receive feedback [58]	P64
SAFe implementation	The acronym stands for Safe Agile Framework. It's a framework containing a set of principles used to implement agile practices at an enterprise scale [59]	P64
Multidisciplinary teams	Teams of employees from different backgrounds and with different knowledge, which can increase their efficiency and creativity when faced with problems [60]	P64
Periodic communication	Providing the latest information about the company and its performance on a regular basis, typically in reports [61]	P64
Workshops	Discussions or collections of exercises on a particular subject, led by people who have experience in it, providing the attendees with an opportunity to gain a lot of knowledge [62]	P69
Market analysis	A thorough analysis of your market within your industry, which helps to avoid risks and make more informed decisions [63]	P69
Design sprint	Intense 5-day processes where teams tackle design problems, working with experts and testing solutions on select users, with the ultimate goal of creating prototypes likely to succeed [64]	P69
Continued on next page		

Technique	Description	Problem addressed
Customer Advisory Board	A group of key customers meant to provide information about the company's reputation and feedback on its offerings [65]	P69
Product discovery	A process of identifying problems, coming up and experimenting with solutions and then testing their usability, improving them throughout the process based on your findings. The process ends when the results of user tests are deemed satisfactory [66]	P69
MoSCoW analysis	The acronym defines the 4 categories of priorities in a project. These are: Must-have, Should-have, Could-have and Won't-have. Assigning features and goals to these groups helps with organising development and prioritising what's important [67]	P69

### **3.7. Chapter summary**

In Chapter 3, the Software Product Management Guide framework was comprehensively presented, elucidating its practical application. The chapter delved into the fundamental components of the framework, namely Problems, Guidelines, Solutions, and Techniques. These components were meticulously crafted based on the insights derived from three distinct Studiums.

In the initial phase, Studium 1, Problems inherent in software product management were methodically identified and prioritized. Subsequently, Studium 2 involved the collection of Solutions, Guidelines, and Techniques directly from practitioners in the field. The culmination of this iterative process was Studium 3, designed to validate the real-world impact of the proposed framework on the software engineering practices adopted by product development teams.

Details pertaining to the research methods employed across all three Studiums will be expounded upon and thoroughly elucidated in the subsequent chapter 4 to 6.

## **4. STUDIUM 1 - PROBLEMS RELATED TO SOFTWARE PRODUCT MANAGEMENT [19]**

The chapter presents studium 1 - research conducted in order to identify key problems related to software product management. This study was published as a publication co-authored by the author of the framework [19].

The first part of the research was to identify specific problems recognized by Software Product Managers and to evaluate the perceived frequency and perceived severity of the selected problems – all from the perspective of software product managers.

The goal of this research is to identify and evaluate the problems of software product management from the perspective of software product managers. The research questions are as follows:

- RQ1: What problems are recognized by software product managers?
- RQ2: What is the perceived frequency of software product management problems by software product managers?
- RQ3: What is the perceived severity of software product management problems by software product managers?

In this research, a software product management problem was defined as a concern related to some topic that affects the work of a software product manager, reduces software product management effectiveness and makes it more difficult to achieve goals. A problem area is a named set of problems originating from a particular area of the company.

As the frameworks described previously present, software product management covers many processes and responsibilities. It connects to many activities carried out in the company. This is why the problems from the perspective of the Software product managers may be related to a wide range of processes, activities and responsibilities of other teams/departments/roles. Many problems may have an impact on a Software product manager's job, and they can be related to core SPM activities or lie beyond them.

The role of the software product manager and the responsibilities attached to this role differ between companies. Problems that affect this role may lay in the direct responsibilities of one product manager, and for another one, they may be external problems that are someone else's responsibility. Many problems have a strong impact on the software product managers job (e.g. technical debt, which slows down the product development process and increases time-to-market), but can not be fixed by this role alone.

As Table 4.1 presents, different research methods were used to answer research questions.

**Table 4.1:** Research methods and data collection techniques used to answer the research questions [19]

ID	Research question	Research method
1	What problems are recognized by software product managers?	systematic literature review, interviews
2	What is the perceived frequency of the software product management problems by the software product managers?	questionnaire
3	What is the perceived severity of the software product management problems by the software product managers?	questionnaire

#### 4.1. Systematic literature review

The first step of studium 1 aimed at answering RQ1 and involved a review of the current literature using the Systematic Literature Review method [68]; [69]). The keywords were derived from RQ1 and multiple independent scientific literature databases were selected as relevant to computer science, together with aggregate databases.

**Table 4.2:** Detailed search criteria of the systematic literature review [19]

Database	Publication type	Discipline	From Year	Scope
Elsevier Science Direct	Review articles and Research articles	-	2013	Full Text
Scopus	Journals, Conference Proceedings, Book Series	-	2013	Title, abstract, and keywords
IEEEExplore	Journals and Magazines	-	2013	Full text
SpringerLink	Articles	Computer Science, Engineering, Business Management	2013	Title, abstract and keywords
Web of Science	-	-	2013	Full text

Inclusion criteria were as follows: conference and journal papers published since 2013 (as ProdBoK® [23] was published in that year); publications in English; original research and review papers. The papers not based on research in software companies were excluded (e.g. in the academic environment) and those related to other papers (e.g. extended versions). Research subjects such as IT hardware, open source software and global software engineering were excluded as laying outside of the scope of the research topic. The inclusion criteria were applied directly in the search engines, while the exclusion criteria were applied during the title and abstract review phases.

Besides the keyword-based search analysis, backwards snowballing [70] was performed to check if any important articles published before 2013 were missed out. The search criteria for snowballing allowed for articles older than 2013 to be included in the results as well as articles without “product manager” or “product owner” in the title. Next, the papers’ titles and abstracts and the duplicates were screened. An iterative process was applied with an independent assessment of each paper’s title and abstract by each researcher, along with a discussion of the discrepancies and a mutual final decision. The iterations were repeated until a unanimous decision for each paper was reached.

For the final data extraction, papers were selected based on the following matching criteria: the “product management,” “product manager,” or “product owner” keywords appear in the title or abstract.

After the first analysis, it turned out that it is not possible to collect from these articles direct problems related to software product management. Because of that, a list of topics mentioned in the context of software product management problems was extracted from each selected paper

and categorized into problem areas using keyword analysis, ground knowledge and industry experience. The topic was defined as a specific task, responsibility, practice, or technique mentioned in the article as a source of software product management problems or a place where such problems might occur, e.g. a lack of continuous integration mentioned as a source of delays in product releasing allowed to identify the topic of “continuous integration.” The problem area was defined as a group of topics that are related to a common portion of the software product management process or framework, e.g. the area of “software development process” which contains topics such as “continuous integration,” “technical debt,” and “software quality.”

Moreover, during the analysis, articles were tagged if they included “product management” in the list of keywords, to understand how many articles were strictly related to the field that was under the research.

During the analysis, it was decided to build a list of problem areas a posteriori from the current literature instead of using an a priori list of areas taken from the product management frameworks to better reflect the current state-of-the-art and reports from industrial practice. The systematic literature review was carried out between November 2018 and February 2019, and involved two researchers. Further, in 2021 backwards and forward snowballing [70] were carried out and included the results in the field of software product management. The list of problem areas identified through the systematic literature review was used later to construct the interview guide and identify the detailed software product management problems from the perspective of software product managers in the next step of the research.

The following search query was used in the systematic literature review: ‘(“product management” OR “product manager” OR “product owner”) AND (software OR IT) AND (problems OR barriers OR limitations OR challenges)’. 5 databases were searched: Elsevier ScienceDirect, Scopus, IEEEExplore, SpringerLink, and Web of Science. Table 2 presents the detailed search criteria.

**Table 4.3:** Number of papers accepted in each step of systematic literature review [19]

Database	Total Found	Screening title and abstract	Removing duplicates	Field fit scanning
Elsevier Science Direct	253	150	150 (0)	20
Scopus	147	81	75 (6)	12
IEEEExplore	231	52	50 (2)	0
SpringerLink	270	71	70 (1)	1
Web of Science	87	46	15 (31)	2
Backward Snowballing	-	-	-	9
Forward Snowballing	-	-	-	3
Total	988	400	361	47

With the aforementioned search criteria, 988 research papers were found. After screening the title and abstract in 3 iterations, a list of 400 relevant papers was obtained. It was reduced to 361 unique papers by removing duplicates. After scanning the full text for the fit to the field of software product management and running backwards and forward snowballing, 47 papers were selected for data extraction to identify the topics of the problems, and possibly also some concrete software product management problems. Table 4.3 presents the number of accepted papers from each database in each step of the systematic literature review. The number in parentheses shows the number of duplicates removed. For the purpose of snowballing, around 400 references in the papers were scanned from the initial systematic literature review result set, 30 sources were checked and 12 of them got accepted. The list of all 47 scanned sources can be found in table 4.4.

**Table 4.4:** List of articles from systematic literature review [19]

ID	Description
S1	Juliana Medeiros, Alexandre Vasconcelos, Carla Silva, Miguel Goulão. Quality of software requirements specification in agile projects: A cross-case analysis of six companies [71]
S2	Sana Ben Nasr, Guillaume Bécan, Mathieu Acher, João Bosco Ferreira Filho, Nicolas Sannier, Benoit Baudry, Jean-Marc Davril. Automated extraction of product comparison matrices from informal product descriptions [72]
S3	George Valença, Carina Alves, Slinger Jansen. Strategies for managing power relationships in software ecosystems [73]
S4	Stefan Wiesner, Sara Nilsson, Klaus-Dieter Thoben. Integrating Requirements Engineering for Different Domains in System Development – Lessons Learnt from Industrial SME Cases [74]
S5	Hiva Alahyari, Tony Gorschek, Richard Berntsson Svensson. An exploratory study of waste in software development organizations using agile or lean approaches: A multiple case study at 14 organizations [75]
S6	Nirnaya Tripathi, Eriks Klotins, Rafael Prikładnicki, Markku Oivo, Leandro Bento Pompermaier, Arun Sojan Kudakacheril, Michael Unterkalmsteiner, Kari Liukkunen, Tony Gorschek. An anatomy of requirements engineering in software startups using multi-vocal literature and case survey [76]
S7	Sezin Gizem Yaman, Myriam Munezero, Jürgen Münch, Fabian Fagerholm, Ossi Syd, Mika Aaltola, Christina Palmu, Tomi Männistö. Introducing continuous experimentation in large software-intensive product and service organizations [77]
S8	Supriya Vasudeva Shrivastava, Urvashi Rathod. A risk management framework for distributed agile projects [78]
S9	Daniel Ståhl, Torvald Mårtensson, Jan Bosch. The continuity of continuous integration: Correlations and consequences [79]
S10	Brian Fitzgerald, Klaas-Jan Stol. Continuous software engineering: A roadmap and agenda [80]
S11	Ahmad Issa, Bumin Hatiboglu, Andreas Bildstein, Thomas Bauernhansl. Industrie 4.0 roadmap: Framework for digital transformation based on the concepts of capability maturity and alignment [81]
S12	Eero Laukkanen, Juha Itkonen, Casper Lassenius. Problems, causes and solutions when adopting continuous delivery—A systematic literature review [82]
S13	Johannes Holvitie, Sherlock A. Licorish, Rodrigo O. Spínola, Sami Hyrynsalmi, Stephen G. MacDonell, Thiago S. Mendes, Jim Buchan, Ville Leppänen. Technical debt and agile software development practices and processes: An industry practitioner survey [83]
S14	Jesse Yli-Huumo, Andrey Maglyas, Kari Smolander. How do software development teams manage technical debt? – An empirical study [84]
S15	Eveliina Lindgren, Jürgen Münch. Raising the odds of success: the current state of experimentation in product development [85]
S16	Antônio Mauricio Pitangueira, Rita Suzana P. Maciel, Márcio Barros. Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature [86]
Continued on next page	



ID	Description
S17	Sami Jantunen, Donald C. Gause. Using a grounded theory approach for exploring software product management challenges [87]
S18	Meghann L. Drury-Grogan. Performance on agile teams: Relating iteration objectives and critical decisions to project management success factors [88]
S19	Andrey Maglyas, Uolevi Nikula, Kari Smolander. What are the roles of software product managers? An empirical investigation [14]
S20	Christof Ebert, Sjaak Brinkkemper. Software product management – An industry evaluation [15]
S21	W.W. Koczkodaj, P. Dymora, M. Mazurek, D. Strzalka. Consistency-driven pairwise comparisons approach to software product management and quality measurement [89]
S22	C. Lin. Data driven product management [90]
S23	V.T. Heikkilä, M. Paasivaara, C. Lasssenius, D. Damian, C. Engblom. Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson [91]
S24	M. Paasivaara, C. Lassenius. Scaling scrum in a large globally distributed organization: A case study [92]
S25	S. Kristinsdottir, M. Larusdottir, Å. Cajander. Responsibilities and challenges of product owners at Spotify – an exploratory case study [93]
S26	A. Morris, R. Setchi, P. Prickett. Product change management and future information architectures [94]
S27	F. Oliveira, A. Goldman, V. Santos. Managing Technical Debt in Software Projects Using Scrum: An Action Research [95]
S28	G. Lucassen, F. Dalpiaz, J.M.V.D. Werf, S. Brinkkemper. Bridging the Twin Peaks – The Case of the Software Industry [96]
S29	J. Bosch. Architecting to Ensure Requirement Relevance: Keynote TwinPeaks Workshop [97]
S30	T.O.A. Lehtinen, R. Virtanen, V.T. Heikkilä, J. Itkonen. Why the development outcome not meet the product owners' expectations? [98]
S31	A. Fabijan, H. Holmström, J. Bosch. Customer feedback and data collection techniques in software R&D: A literature review [99]
S32	S. Jantunen, K. Hietaranta, D.C. Gause. Adjusting to increasing product management problems: Challenges and improvement proposals in one software company [17]
S33	M. Komssi, M. Kauppinen, H. Töyhönen et al. Roadmapping problems in practice: value creation from the perspective of the customers [100]
S34	Andrey Maglyas, Uolevi Nikula, Kari Smolander, Samuel Fricker. Core Software Product Management Activities [16]
S35	Farnaz Fotrousi, Katayoun Izadyan, Samuel Fricker. Analytics for Product Planning: In-Depth Interview Study with SaaS Product Managers [101]
Continued on next page	

ID	Description
S36	Sebastian Barney, Aybūke Aurum, Claes Wohlin. A product management challenge: Creating software product value through requirements selection [102]
S37	Christof Ebert. The impacts of software product management [3]
S38	Gabriel Steinhardt. The Product Manager's Toolkit® [103]
S39	Christof Ebert. Software Product Management [34]
S40	Hans-Bernd Kittlaus, Samuel A. Fricker. Software Product Management [2]
S41	Andrey Maglyas, Uolevi Nikula, Kari Smolander. Lean Solutions to Software Product Management Problems [13]
S42	Andrey Maglyas, Uolevi Nikula, Kari Smolander. Comparison of Software Product Management Practices in SMEs and Large Enterprises [104]
S43	Andrey Maglyas, Uolevi Nikula, Kari Smolander. "What do practitioners mean when they talk about product management?" [105]
S44	Andrey Maglyas, Samuel Fricker. The Preliminary Results from the Software Product Management State-of-Practice Survey [106]
S45	I. van de Weerd, S. Brinkkemper, R. Nieuwenhuis, J.M. Versendaal, A. Bijlsma. A reference framework for software product management [28]
S46	E. Kim. User-Centered Design Roadmapping: Anchoring Roadmapping in Customer Value Before Technology Selection [107]
S47	Timo Wagenblatt. Software Product Management: Finding the Right Balance for YourProduct Inc. [32]
S48	Andrey Saltan, Slinger Jansen, Kari Smolander. Decision-making in Software Product Management: Identifying Research Directions from Practice [108]

It was found out that the reviewed papers named many areas and topics where software product management problems might occur. 7 problem areas were coded (A1 to A7) to group similar topics.

The list of problem areas and topics shown in Table 4.5 and examples of the problems identified from the literature presented in Table 4.6 are the final results of literature review.

**Table 4.5:** Problem areas and topics extracted from the analyzed literature [19]

Id	Problem area	Topics
A1	Requirements engineering	Requirements management, prioritization, stakeholders management, product roadmap, user involvement
A2	Software development process	Technical debt, continuous integration, release cycle speed / time-to-market, software quality, system architecture, split testing, data collection
A3	Team	Unqualified people, motivation, communication
A4	Product knowledge management	Product description, sharing knowledge, documentation
A5	Organizational strategic –	Strategy, business model, market research, price management), trust in team, partnership management
A6	Organizational operational –	Tools, responsibilities, distributed teams, continuous experimentation, decision-making process, agile, time, budget
A7	User research	User centered design, customer feedback, user behavioral data, business intelligence, product design )

The research finds out that 4 topics that occur the most often as the problems in the literature about software product management are related to responsibilities, stakeholders management, strategy, and decision-making process. Examples of problems from the literature assigned to these topics can be found below.

Responsibilities: “Overlapping responsibilities, an overwhelming volume of activities.”; “We often find that product management is more of a role label than an effective function that leads the product.”; “Challenges like too many responsibilities and little authority.”

Stakeholders management: “Talked about the stakeholder side as being one of the most challenging parts of their role.”; “The alignment of product, project and business decisions is a major problem in the software industry.”; “Ambiguous role definition, imbalanced relationships with other departments.”

Strategy: “Disadvantage of a feature-driven mind-set.”; “Short-term thinking.”; “Lack of strategy and unclear strategy and roadmaps with unclear dependencies and fuzzy technical requirements and impacts.”

Decision-making process “This randomness of decision-making was typically accompanied by severe communication problems between IT and the business units.”; “No standardized processes across the company with a slow and cumbersome decision-making process and many individual ad-hoc agreements.”; “Problems related to making strategic and tactical decisions about a particular product without permission from the higher management. In some cases decision-making was considered as the main characteristic of a product manager.”

**Table 4.6:** Examples of the problems identified from the literature [19]

L1. The product manager is a person who can influence what happens only by persuasion and motivation.	[14]
L2. Responsibility and authority are not shared between people.	[14]
L3. Collaboration with other departments including marketing, sales, and development departments.	[14]
L4. Making strategic and tactical decisions about a particular product without permission from the higher management. In some cases decision-making was considered as the main characteristic of a product manager.	[14]
L5. Limited authority of the product manager.	[14]
L6. I get an idea about our product but I do not know how many people are interested in that.	[14]
L7. PM responsible for the communication function only. He or she was a facilitator between all departments including marketing, development, sales, and support, and was responsible for taking part in solving their problem.	[14]
L8. A dysfunctional organization with unclear responsibilities and silo work which results in continuously changing focus and schedules.	[15]
L9. The problem is not so much that the stakeholders are not known, but that the interdependencies, roles and expectations to requirements are unclear.	[15]
L10. A number of challenging issues were found, including communication gaps between marketing and development, selecting the right level of process support, basing the release plan on uncertain estimates, and managing the constant flow of requirements.	[15]
L11. Product managers are mostly subordinate to marketing and play an administrative role, such as maintaining technical roadmaps.	[15]
L12. Insufficient requirements engineering is typically the first sign of product management failure. Even as they try to specify requirements, they are not developing requirements with other stakeholders.	[15]
L13. A weak perception of the role and its incumbents by senior management, which in turn would not strengthen the role.	[15]
L14. The difficulties with misunderstanding needs, changing and creeping requirements, missed deadlines and budgetary commitments and failing business opportunities.	[15]
L15. No standardized processes across the company with a slow and cumbersome decision-making process and many individual ad hoc agreements.	[15]
L16. Insufficient requirements which are often just collections of what had been heard at customer visits and other such events, but not mapped to value creation and business cases.	[15]
L17. Lack of strategy and unclear strategy and roadmaps with unclear dependencies and fuzzy technical requirements and impacts.	[15]
L18. It is difficult to motivate a team that does not know for whom it is working and that lacks a purpose.	[93]
L19. Communication was also mentioned as being one of the Product Owners' main challenges.	[93]
L20. Long release cycles, short-term thinking, and a lack of constant collaboration between organizations and customers.	[16]
L21. No metrics for evaluating work in all of the organizations, the product managers responsible for the whole product had no key performance indicators (KPIs).	[13]
L22. Collaboration between organizations and customers.	[13]
L23. Short-term thinking.	[13]
L24. Trying to change instantly.	[13]
L25. Strategic and tactical planning are tightly coupled together, but the main problem is the balance between them.	[104]
L26. In SMEs, most decisions were made by top management and product managers acted as advisors and facilitators in solving problems arising between departments.	[104]
L27. Issues related to roadmapping, and especially to aligning problems with business goals.	[105]
L28. The main problem with these responses was that after attaching the label mini-CEO to the product manager, there were no explanations of what the mini-CEO should do.	[105]

## **4.2. Interviews**

The second step aimed at identifying software product management problems with qualitative research. To fill the research gap in the literature, a series of semi-structured moderated interviews with industry practitioners were conducted to identify the problems in software product management from the perspective of the actual software product managers, following the guidelines by Hove and Anda[109]. Interviews were chosen as the research method to facilitate the free identification of problems and keep the scope and depth of research data as open as possible. Semi-structural approach was employed to ensure the coverage of all problem areas from the literature but also to allow the interviewees to mention problems whenever it was convenient for them.

Participants for the interviews were recruited using purposive sampling [110]. The interviewees were qualified by their experience in software product management and the size of the company. 5 years of experience in software product management was assumed to be a sufficient level of expertise regarding software product management problems. Experts from companies of various sizes were invited to cover the specific problems related to the role of a product manager which differs by company size, as shown in previous research [30]. Interviewees were invited using personal contacts and various interest groups on social media. The interviews were carried out in the form of face-to-face or online meetings using Skype and lasted 1 to 2 hours. Interview guide followed all of the problem areas from the literature and used the topics of each problem area as prompts. The experts were provided the interview guide in advance, however not everyone read it before the meeting. The interviews were recorded for further analysis. Questions about problem areas were randomized during the interview in order to minimize the impact of the expert's fatigue. The interviews were carried out from late January 2019 to April 2019.

A list of software product management problems was extracted separately from each interview preserving the original interviewees' statements. Interviews were stopped when 10 problems that were mentioned in more than half of the interviews were identified. Inductive coding was carried out, which is also known as open coding. Inductive coding means starting from scratch and creating codes based on the qualitative data itself. All codes arose directly from the interviewee's responses. After the interview, the work was to convert complex sentences into codes (simplified sentences that are names of problems). When creating the codes, syntax in order was taken into account to maintain a similar meaning. To ensure the reliability of the coding, interviews recordings were double-listened, to make sure problems were correctly identified and described by the interviewee. Additionally, the extracted and coded problems were also double-verified.

Then, the problems were screened for duplicates and identical problems mentioned in different interviews were merged. Combining at this stage consisted of removing obvious duplicates, i.e. the same codes in different interviews (repeated problem names). The problems were also assigned to the problem areas based on the interview sections the problems were mentioned in as well as a post-interview keyword analysis of the problem statements. Many problems were assigned to multiple areas to reflect the fact that the problems span, stem from or affect several aspects of the software product management activities and lifecycle. This recognizes the complexity of software product management which is consistent with other research [13] [34].

Finally, all problems were analyzed again for their similarity to allow for further merging between different interviewees. The second round of merging consisted of combining different codes which were found to essentially describe the same problem. The final problem was

assigned to all of the areas of the merged problems. The resulting list of problems contained only the problems that couldn't be further merged without losing their meaningful details.

The list of problems was further used to select the problems for the evaluation of their perceived frequency and severity in the next step. The examples below show how the codes are based on the extracted data.

Example 1. Code: Lack of user analytics

- No access to user behaviour data – enterprise customers do not allow it to be tracked,
- Lack of data about user behaviour,
- Lack of data analytics,
- No data about product usage,
- Lack of data setup,
- Lack of data analytics on user behaviour.

Example 2. Code: No company strategy

- Company strategy not defined, PM does not know what goals and projects he should focus on,
- Lack of strategy,
- Strategy not defined,
- Wrong strategy – detailed plan without high-level strategy,
- No clear strategy, no clear priority,
- Lack of a visioner on board.
- If you lack a product strategy then you can not prioritize what you are going to do,
- Stakeholders management – they have different visions and goals

Example 3. Code: Teams are not agile, they just follow rules and do not use experimentation and a learning process

- Teams are not agile and do not understand what it means,
- People do not understand agile, the need for experimentation and the learning process,
- Company can not stand failures, treats overthrowing the hypothesis as the failure, does not understand agile and lean philosophy, blames people for 'failures',
- People do not understand Agile – they just follow Scrum rules,
- People do not measure because they are scared of failure,
- Lack of developer's willingness to be agile,
- People in scrum behave like automatons, they forget about empiricism, they must have design and precise descriptions to start work on the task,
- Scrum – bragging about small things, celebrating small things too frequently – because of that we lose sight of the big picture and long-term goals.

The software product management problems were identified with a set of 10 interviews with practising software product managers following a predefined interview guide. Table 4.7 shows the characteristics of the interviewed experts. Company size is given as the approximate number of employees. One of the experts (I3) had less than 5 years of experience in 2019 but was eligible for the interview due 3 years of additional experience working as a project manager and agile coach in IT companies that build and develop their own products, where he also cooperated with roles responsible for software product management. Additionally in 2021, the same expert, having 4 years of experience and working as a Lead Product Manager, validated the results again and confirmed their validity.

**Table 4.7:** Characteristics of interviewed software product managers

Id	Company name	Company size	Position	SPM experience [yrs]
I1	Atlassian	3500	Group Product Manager	15
I2	STX Next	300	Product Owner	5
I3	AirHelp	600	Product Manager	2
I4	Dynatrace	2000	Senior Technical Product Manager	12
I5	Sparandmortal	30	Product Manager	10
I6	CAS360	120	Head of Product	7
I7	SentiOne	80	Product Manager	8
I8	Vendavo	250	Product Management Consultant, VP of Product Management	12
I9	Lufthansa	650	Senior Product Owner	8
I10	AirHelp	600	Head of Product	8

In total, the interviewees provided 237 problems related to software product management. The same problems (codes) that were indicated in different interviews were merged, resulting in a list of 163 software product management problems. Similar problems were further merged, resulting in a final list of 95 unique problems, which is available in the supplementary data in the online version of this article. The most common problem was indicated in 8 interviews. 27 problems were stated in at least 3 interviews. This threshold was decided to have a feasible number of problems to study in this research. It may be decided to investigate other, more rarely mentioned problems in the future. Additionally, the entire list of 95 problems were studied and any new problem areas outside of the selected 27 problems wasn't found, which shows that the coverage of the software product management scope is satisfactory.

**Table 4.8:** Software product management problems identified in at least 3 of 10 interviews [19].

ID	Problem name	Problem areas	No. of interviews
P13	The team's lack of motivation due to a lack of understanding of why they are doing this – no commitment to achieving goals	Team	8
P9	Technical debt	Software development process	7
P10	Lack of user analytics data	Software development process, User Research, Requirements Engineering	7
P16	Lack of synchronization between product management and marketing and sales units	Product knowledge management	7
P1	No company strategy	Requirements Engineering, User research, Organizational – strategic	6

Continued on next page

ID	Problem name	Problem areas	No. of interviews
P3	Wrong data analytics setup	Requirements Engineering, Use research, Software development process	6
P6	Lack of continuous integration and delivery (impact on 'time to market')	Requirements Engineering, Software development process	6
P18	Lack of market research, no understanding of business and trends in the industry	Organizational – strategic	6
P25	Limited access to users in order to do research	User research, Requirements Engineering	6
P51	Roadmap focused on features instead of goals and business value	User Research, Software development process, Operational – strategy, Requirements Engineering, Team, Product Knowledge Management	6
P19	Teams are not Agile, they just follow rules and do not use experimentation and a learning process	Organizational – operational, Team	5
P26	Lack of support for research (no resources allocated to the team)	User research	5
P44	Teams are not autonomous and self-organized (difficult to organize, dispersed both responsibility and decision-making)	Organizational – operational, Team	5
P48	Lack of user research	User Research	5
P43	Lack of trust to the product team (micro-management)	Organizational – strategic, Requirement Engineering, Organizational – operational, Team	4
P69	Balancing between reactive and proactive work. When comparing hypotheses with facts, hypotheses lose in value to facts (such as clients' requests, bugs).	Requirements Engineering, User research, Operational – strategy	4
P74	Determining the true value of the product that the customer needs	Operational – strategy, Requirements Engineering, Software development process	4
P82	PM role not clearly defined and communicated in the organization (what the role is about, what the responsibilities and objectives are, decisiveness)	Team, Organizational – operational, Product Knowledge Management, Operational – strategy	4
P8	Different expectations about product management communication per stakeholder (high/low level)	Requirements Engineering, Organizational – strategic, Product Knowledge Management	3
P12	Low software quality	Software development process	3
P27	Unqualified team members (individuals)	Team	3
P35	Strategy and priorities are changing frequently	Organizational – strategic	3
P41	High expectations from external partners, which are not possible to be met	Organizational – strategic	3
P56	Price management is always an experimentation burdened with risk	Organizational – strategic	3
P64	Working in silos (problem with communication, synchronizaton between teams)	Product Knowledge Management, Organizational – operational, Team	3
P78	Lack of skills to use and analyze the data	User Research	3
P85	Lack of automatic testing	Software development process	3

As already mentioned in chapter 2, Maglyas et al. provided a list of the 5 main problems



related to software product management [13]. In the research, they studied 13 organizations to gain an understanding of how software product management practices were adopted. Table 4.9 provides an insightful mapping of the 27 problems identified in this research, to the five problems recognized by Maglyas et al. in 2012. While identified in this study problems may not perfectly align with those of Maglyas et al., a comprehensive review of the problem descriptions in the article reveals shared thematic areas.

**Table 4.9:** Software product management problems found in this research mapped onto problems of Maglyas et al. using recognized common themes [19].

Rnk	Problem name	Maglyas et al. 2012	Common theme
1	P74. Determining the true value of the product that the customer needs	Problem 3: Collaboration between organization and Customers	customer focus
2	P35. Strategy and priorities are changing frequently	Problem 4: Short-term thinking	strategy
3	P9. Technical debt	-	-
4	P64. Working in silos (problem with communication, synchronization between teams)	Problem 1. Long Release Cycle	silos/units
5	P69. Balancing between reactive and proactive work	-	-
6	P26. Lack of support for research (no resources allocated to the team)	Problem 3. Collaboration between Organization and Customer	resources
7	P85. Lack of automated testing	-	-
8	P82. Product Manager role not clearly defined and communicated in the organization (what the role is about, what the responsibilities and objectives are, decisiveness)	-	-
9	P48. Lack of user research	Problem 3. Collaboration between Organization and Customer	research
10	P51. Roadmap focused on features instead of goals and business value	Problem 2: No metrics for evaluating work	metrics, business value
11	P18. Lack of market research, no understanding of the business and trends in the industry	Problem 4. Short-term thinking	trends
12	P8. Different expectations about product management communication per stakeholder (high/low level)	-	-
13	P6. Lack of continuous integration and delivery (impact on 'time to market')	-	-
14	P25. Limited access to users in order to do research	-	-
15	P16. Lack of synchronization between product management and marketing and sales units	Problem 1. Long Release Cycle	units
16	P1. No company strategy	Problem 4: Short-term thinking	strategy
17	P44. Teams are not autonomous and self-organized (difficult to organize, dispersed both responsibility and decision-making)	-	-

Continued on next page

Rnk	Problem name	Maglyas et al. 2012	Common theme
18	P10. Lack of user analytics data	-	-
19	P19. Teams are not Agile, they just follow rules and do not use experimentation and a learning process	-	-
20	P13. The team's lack of motivation due to a lack of understanding of why they are doing things – no commitment to achieve goals	-	-
21	P41. High expectations from external partners, which are not possible to be met	-	-
22	P3. Wrong data analytics setup	-	-
23	P12. Low software quality	-	-
24	P78. Lack of skills to use and analyze the data	-	-
25	P43. Lack of trust in the product team (micro-management)	-	-
26	P27. Unqualified team members (individuals))	-	-
27	P56. Price management is always experimentation burdened with risk	-	-

For instance, Maglyas et al.'s Problem 3, addressing collaboration between the organization and the customer, was mapped onto three of problems (P74, P26, and P48). This mapping underscores the enduring nature of the challenge, as evidenced by the consistently high perceived frequency and severity of five identified problems. Maglyas et al.'s Problem 4, focusing on short-term thinking, was found to resonate with three problems identified in this study, suggesting room for enhancement in strategy-related aspects of software product management.

Problem 1 from Maglyas et al., centered on long release cycles, correlated with two problems. This alignment hints at opportunities for improvement in structuring teams using different methods to foster collaboration and efficiency, thereby avoiding silos. Maglyas et al.'s Problem 2, emphasizing the absence of metrics for evaluating work, was mapped to one problem. While this research suggests a lower perceived frequency of this problem, it may indicate that companies have recognized the importance of measuring software product management performance for overall success since 2012.

Interestingly, the only problem from Maglyas et al. not covered in this study is Problem 5: Trying to change instantly. This absence may signify that organizations have learned effective ways to introduce software product management processes over time, reducing the perceived challenges.

This research also identified 18 unique problems that could not be matched with any problems identified by Maglyas et al. Notably, technology-related issues (P9, P85, P6, P12) and competency-related challenges may have emerged due to the rapid development of the IT market. Stakeholder management challenges may have intensified due to the expanding scope of software product management and the rising number and diversity of stakeholders involved. The emergence of big data has brought forth problems in user and data analytics, areas that were not as prominent in 2012 but have gained significance today.

### 4.3. Survey

In the next step of the study, the 27 software product management problems identified previously (shown in table 3.1), were evaluated with the survey.

To establish evaluation criteria for the problems, recognized standards commonly used in the realm of risk assessment were applied. One widely used framework for risk assessment is provided by the U.S. Environmental Protection Agency (EPA), which defines risk as the combination of the probability or likelihood of an adverse effect and the severity of that effect. This framework typically involves four steps: hazard identification, dose-response assessment, exposure assessment, and risk characterization. Measurement of risk involves quantifying both the probability and magnitude of potential harm. Various methods and metrics can be used depending on the specific risk being assessed, including probabilistic modeling, statistical analysis, and expert judgment. Similar to problems, while the definition and quantification of risk may vary depending on the situation, it typically pertains to the likelihood of an adverse event happening and the potential consequences it may entail [111].

So in the survey frequency of occurrence and severity of the problems were used as variables for further research. As it was not possible to analyse whole population, proxy measure was analysed based on responses from software product managers.

The survey was divided into three parts. In the first part, the demographic questions were asked about the respondent's country, experience in IT (in years), roles in IT that the experts had held at any point in their career, experience as a software product manager (in years), name of the role when responsible for software product management, and size of the company when working as a software product manager.

In the second part, for each problem the respondents were presented with two statements:

a) I experience this problem in my work as a product manager b) This problem has a significant impact on my work as a product manager

Respondents were asked to agree or disagree with these statements in a Likert 5-point response format (strongly disagree, disagree, neutral, agree, strongly agree) [112]. Statement (a) was used to evaluate the perceived frequency of the problem, whereas statement (b) was used to evaluate the perceived severity of the problem.

Perceived frequency was defined as the percentage of the product managers that agree that they experienced a given problem in their work and calculated it as the total percentage of "agree" and "strongly agree" responses to the above mentioned question (a) on that problem. It is important to note that this definition is based on the respondents' opinions, which is typical of qualitative research.

The quantitative probability of problem occurrence was not referred, which could be biased by non-random sampling. Similarly, the perceived severity was defined as the percentage of the product managers that agree that a given problem had a significant impact on their work and calculated as the total percentage of "agree" and "strongly agree" responses to the above mentioned question (b) on that problem. Again, an opinion based qualitative definition was implemented instead of any quantitative definition based on measurable loss value due to the occurrence of a problem. This approach is used in quantitative research in software engineering [113].

In total, 333 respondents started the survey. It was completed by 89 respondents. 70% of respondents were from Poland, and 30% from other countries (USA, UK, Germany, Australia, Brazil, Portugal, Canada, Denmark, India, and The Netherlands). Their professional experience in

IT was 9 years on average (median: 8 years). 88.7% of the respondents were working as a product manager at the time of the survey or in the past. The respondents stated that their experience as a software product manager was 3.7 years on average (median: 3 years).

The distribution of the names of the respondents' roles responsible for software product management is shown in table 4.10. Multiple answers were allowed. Their role names were mostly: Product Owner, Product Manager, and Head of Product. 11 respondents provided some other names of their roles which were: Product Designer, Product Owner Trainee, Creative Technologist, Product Specialist, Senior Technical Project Manager, Technical Product Manager, Project Manager, Director of Product Management, Product Analyst, Product Lead, and UX. This demonstrates that the naming of the role responsible for software product management may be different, but the most common job titles are Product Manager and Product Owner.

**Table 4.10:** Distribution of names of respondents' software product management roles

What is/was the name of the role where you are/were responsible for software product management?	Number of respondents	% of respondents
Product Owner	58	65.1%
Product Manager	51	57.3%
Head of Product	10	11.2%
Software Product Manager	5	5.6%
Chief Product Officer	3	3.3%
Other name of the role responsible for software product management	11	12.4%

The distribution of the roles in IT held by the survey respondents anytime in their career is given in table 4.11. Multiple roles could be selected. 9 respondents added some other roles which were: UX, Technical Writer, Support, Project Manager, Head of Product Management, Director of Product Management, Product Specialist, Support helpdesk, and Business Analyst. These results show that people with experience in the roles of a Project Manager, Analyst, Programmer, Designer, Tester or Product Owner most often become Software Product Managers.

**Table 4.11:** Distribution of respondents' roles in IT anytime in their career

Roles in IT you have held at any time in your career?	Number of respondents	% of respondents
Product Manager	79	88.7%
Project Manager	40	44.9%
Analyst	29	32.0%
Programmer	21	23.6%
Designer	20	20.2%
Scrum Master	17	19.0%
Tester	14	15.7%
Product Owner	12	13.5%
Architect	5	5.6%
DevOps	0	0.0%
Validation Engineer	0	0.0%
Other	9	10.1%

The distribution of the size of the respondents' companies is presented in table 4.12. Multiple answers were allowed to take into account respondents' software product management experience in several companies over their careers. 55% of the respondents came from enterprises of more than 250 employees. This is consistent with previous research, which showed that dedicated software product managers were being hired when the company started to grow [30].

Table 4.13 presents the survey results - the evaluation of the perceived frequency and

**Table 4.12:** Distribution of respondents' company sizes

Company size	Number of respondents	% of respondents
1–10	17	19.10%
11–50	29	13.58%
51–250	33	37.07%
>250	49	55.05%

perceived severity of the 27 software product management problems in survey. The perceived frequency and perceived severity of each problem were calculated as the percentage in agreement, which is the total percentage of “agree” and “strongly agree” responses to that problem.

**Table 4.13:** Evaluation of the perceived frequency and perceived severity of the selected software product management problems

Rnk	Id	Problem name	Frequency [%]	Severity [%]
1	P74	Determining the true value of the product that the customer needs	72.7	69.3
2	P35	Strategy and priorities are changing frequently	71.6	70.5
3	P9	Technical debt	70.5	64.8
4	P64	Working in silos (problem with communication, synchronization between teams)	69.3	65.9
5	P69	Balancing between reactive and proactive work. When comparing hypotheses with facts, hypotheses lose in value to facts (such as clients' requests, bugs). Managing requirements instead of identifying problems and opportunities, seeking innovation.	69.3	63.6
6	P26	Lack of support for research (no resources allocated to the team)	65.9	63.6
7	P85	Lack of automated testing	65.9	62.5
8	P82	Product Manager role not clearly defined and communicated in the organization (what the role is about, what the responsibilities and objectives are, decisiveness)	63.6	58.0
9	P48	Lack of user research	61.4	64.8
10	P51	Roadmap focused on features instead of goals and business value	61.4	59.1
11	P18	Lack of market research, no understanding of business and trends in the industry	55.7	55.7
12	P8	Different expectations about product management communication per stakeholder (high/low level)	55.7	52.3
13	P6	Lack of continuous integration and delivery (impact on 'time to market')	55.7	48.9
14	P25	Limited access to users in order to do research	54.5	59.1
15	P16	Lack of synchronization between product management and marketing and sales units	54.5	52.3
16	P1	No company strategy	53.4	61.4
17	P44	Teams are not autonomous and self-organized (difficult to organize, dispersed both responsibility and decision-making)	53.4	58.0
18	P10	Lack of user analytics data	48.9	54.5
19	P19	Teams are not Agile, they just follow rules and do not use experimentation and a learning process	47.7	51.1
20	P13	The team's lack of motivation due to a lack of understanding of why they are doing things – no commitment to achieve goals	47.7	46.6
21	P41	High expectations from external partners, which are not possible to be met	47.7	45.5
22	P3	Wrong data analytics setup	44.3	44.3
23	P12	Low software quality	43.2	45.5
24	P78	Lack of skills to use and analyze the data	43.2	43.2
25	P43	Lack of trust in the product team (micro-management)	42.0	51.1
26	P27	Unqualified team members (individuals)	42.0	50.0
27	P56	Price management is always experimentation burdened with risk	36.4	29.5

The table is ordered first by perceived frequency and second by perceived severity. The

analysis revealed that the data received on the most severe and frequent problems is consistent with the results of the literature review, but it also identified other areas where problems occur and impact software product managers. Some of the identified problems in this research spanned beyond the software product management process itself, but they all affect the work of software product managers [19].

The 5 most common problems from the perspective of software product managers are:

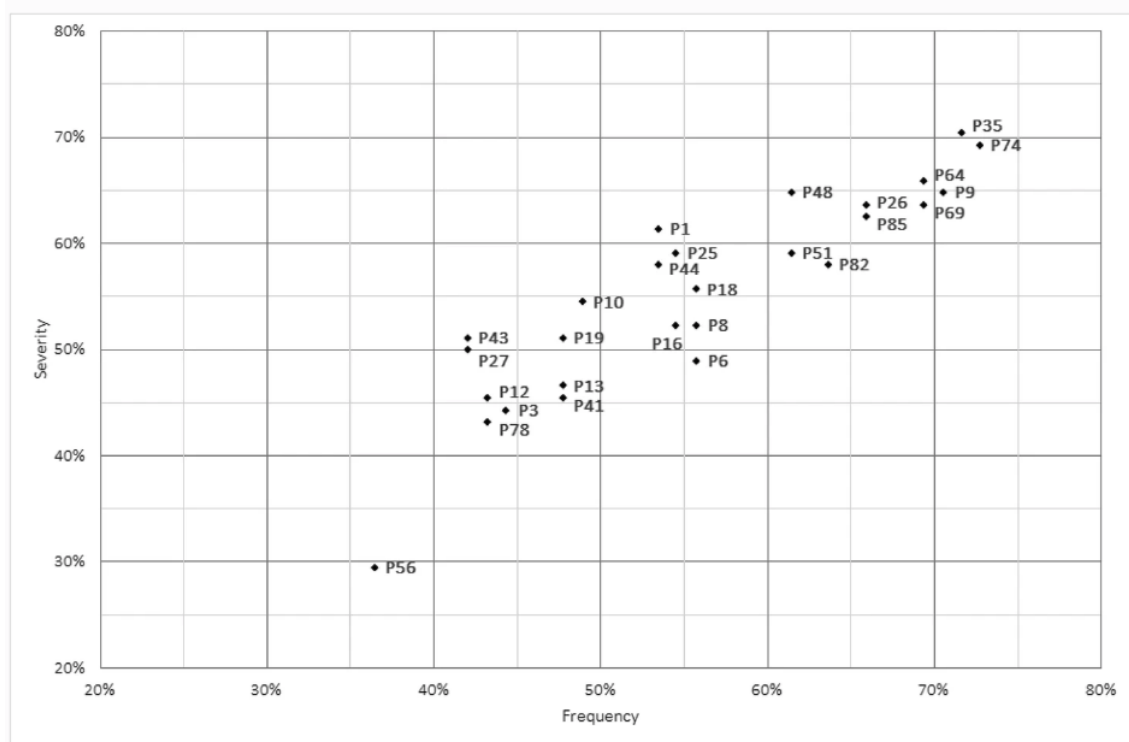
- **P74.** Determining the true value of the product that the customer needs - Product managers have to run research and work iteratively with teams to understand the customer needs, scaling opportunities and customer willingness to pay for the product.
- **P35.** Strategy and priorities are changing frequently - One of the organisational problems that is affecting Product Managers and other roles widely. When strategy is changing frequently, Product Managers and their teams struggle with prioritisation, don't see the long term picture and can't achieve outcomes, as direction is changing too often.
- **P9.** Technical debt - Challenge that is slowing down product development process and makes it even more difficult to prioritise the Product Roadmap. Technical debt is hard to measure, but needs to be watched and prioritise by teams wisely. Good cooperation between Product&Engineering is required to either minimise, pay it off completely or accept the risk.
- **P64.** Working in silos (problem with communication, synchronisation between teams) - Initiatives run across different departments require more discipline to align teams around common goals, synchronise them and organise communication that will rather support team than slowing them down.
- **P69.** Balancing between reactive and proactive work. - For mature products it is always a challenge to prioritise research, innovation and new value for customer against queue of bugs, escalation, maintenance related work.

The identified problems show that Software Product Managers also struggle with gathering data, requirements management, and accessing customers in order to develop valuable solutions. Some of the problems are specific to Software Product Manager role, but a significant part of them are organisational challenges that impact other roles as well [19]

Figure 4.1 shows the same data plotted in the frequency-severity space typical to the risk matrices used commonly in risk management in order to indicate problems with the highest perceived frequency and perceived severity at the same time [114] [115].

The acquired distributions were additionally examined to assess the impact of various confounding variables, including the respondent's role, country, company size, and level of software product management experience. To conduct this analysis, the Knime tool and the Mann-Whitney U-test were employed.

To test for the role of the respondent, the sample was divided into two groups: Product Owner role and non-Product Owner role. The non-Product Owner role covers Product Manager, Software Product Manager, Head of Product, Chief Product Officer and other roles (see Table 8). As a result, observed only one statistically significant difference was observed ( $p=0.028$  for perceived frequency and  $p=0.041$  for perceived severity,  $\alpha=0.05$ ) for problem P19 "Teams are not Agile, they just follow rules and do not use experimentation and a learning process," which was evaluated as more frequent and severe by non-Product Owners than by Product Owners. This might result from the fact that the Product Owner is an Agile (Scrum) role in itself, thus Product Owners work with Agile and face the problem on non-Agile teams more rarely and less severely.



**Figure 4.1:** Evaluation of the software product management problems in the frequency-severity space

To test the respondents' country, sample was divided into two groups: Poland and Rest of World. It was observed that two problems were perceived as more frequent in Poland than in Rest of World: P10 "Lack of user analytics data" ( $p=0.045$ ,  $\alpha=0.05$ ) and P78 "Lack of skills to use and analyze the data" ( $p=0.024$ ,  $\alpha=0.05$ ). The same problem P78 was also perceived as more severe in Poland than in Rest of World ( $p=0.025$ ,  $\alpha=0.05$ ). This result might indicate lower data analytics competencies across software product managers in Poland compared to the rest of the world.

To test the company size, the sample was divided into two groups: large companies and micro to medium companies. It was also observed that two problems were perceived as significantly more frequent in large companies than in micro to medium companies: P74 "Determining the true value of the product that the customer needs" ( $p=0.033$ ,  $\alpha=0.05$ ) and P64 "Working in silos (problem with communication, synchronization between teams)" ( $p=0.045$ ,  $\alpha=0.05$ ). Any statistically significant differences in the perceived severity between the two groups based on company size were not observed. These two problems might indicate that the sheer size and complex internal structure of a large company makes software product management more problematic.

To test software product management experience, the sample was divided into two groups: with less than 3 years of SPM experience, and 3 or more years of SPM experience. Two problems exhibited statistically significant differences in the evaluation of perceived frequency between the two groups. The problem P10 "Lack of user analytics data" was perceived as more frequent by more experienced software product managers, whereas the problem P41 "High expectations from external partners, which are not possible to be met" was perceived as more frequent by less experienced software product managers. The difference for the problem P10 might indicate that more experienced managers want to use hard data in their decision-making more often, which is not always available. The difference for P41 indicates that less experienced managers struggle more often with high expectations, which they cannot meet possibly due to their inexperience.

#### **4.4. Summary**

In total, 99 software product managers actively participated in this study, aimed at identifying the primary challenges within software product management.

The systematic literature review conducted revealed a significant research gap: the problems associated with software product management and potential strategies were not adequately reviewed and summarized.

Conducting interviews and gathering data from software product managers as part of the survey enabled the creation of the first component of the proposed Software Product Management Guide - a comprehensive list of **27 problems**, categorized by perceived frequency and severity. This list will serve as the foundation for the subsequent phase of the research, referred to as Studium 2.



## **5. STUDIUM 2 - STRATEGIES TO MAIN PROBLEMS RELATED TO SOFTWARE PRODUCT MANAGEMENT**

The chapter presents studium 2 - research conducted in order to identify and evaluate the solutions and techniques to the most relevant software product management problems. This study was also published as a publication co-authored by the author of the framework [20] and this chapter is heavily based on this paper.

For the purpose of this study, three research questions were posed:

**RQ1.** How do product managers solve the identified problems in their work?

**RQ2.** To what extent reported solutions to the most common problems are comprehensible?

**RQ3.** What is the effectiveness of these solutions in solving the problem?

### **5.1. Study design**

The aim of this study was to identify and evaluate a set of solutions to the most important problems that arise in the work of product managers. A mix-method approach was used for this purpose. The sequential exploratory strategy [116] was applied, which additionally allowed for increased accuracy and reliability with data triangulation. To answer RQ1, i.e. to identify solutions used by practitioners in their daily work, the focus group method was applied, as it is reported to be suitable for gathering experience from expert groups [116]. Whereas a questionnaire survey was used to evaluate and select the most comprehensible (i.e. clear and easy to understand) and effective of the proposed solutions (RQ2, RQ3). This is consistent with Kontio et al. guidelines, which indicate that focus groups are typically not the only research method used in a study [116].

### **5.2. Focus groups study**

The survey using the focus group method was conducted in accordance with the guidelines proposed by Kontio et al. [116], which were developed on the basis of feedback and experiences from software engineering practitioners. These guidelines have been adopted as a common approach to conducting research using this method in the field of software engineering.

In the remainder of this subsection, the next steps are described in accordance with the guidelines, i.e. planning the research, designing focus groups and conducting the focus group sessions. While the last step, analyzing the data and reporting the results, is presented in section 3.5.

#### **5.2.1. Planning the Research**

The purpose of the focus group study was to identify solutions to common problems that product managers face in their daily work. In the previous study, 27 such problems [19] were recognized. In order to conduct a productive study, it was decided to divide the problems into categories, where each one would be the subject of separate focus groups. A posteriori

classification was used, assigning 2 to 4 problems to one of 8 categories.

Table 5.1 shows the list of categories with descriptions and assigned problems. For consistency, problems are numbered as in the source article [19].

**Table 5.1:** List of categories and assigned problems

Category	Problems
<b>1. Strategy</b> In this area, the Product Manager cooperates with the Management Board in creating the company's strategy and adjusting the strategy for product development to it.	P1. No company strategy. P35. Strategy and priorities are changing frequently. P18. Lack of market research, no understanding of business and trends in the industry.
<b>2. Providing value and searching for business potential</b> In this area, the Product Manager explores the market, and target groups, conducts research to understand users' needs and finds business potential.	P74. Determining the true value of the product that the customer needs. P69. Balancing between reactive and proactive work.
<b>3. User research</b> In this area, the Product Manager conducts research with users in order to have a good understanding of their problems, expectations and experiences with the product.	P48. Lack of user research. P25. Limited access to users in order to do research P26. Lack of support for research (no resources allocated to the team).
<b>4. Technical challenges</b> In this area, the Product Manager works closely with production teams and is directly affected by technical problems encountered by the team.	P9. Technical debt. P12. Low software quality. P85. Lack of automatic testing. P6. Lack of continuous integration and delivery (impact on 'time to market').
<b>5. Data analytics</b> In this area, the Product Manager implements analytical tools and uses them to make decisions about product development.	P3. Wrong data analytics setup. P78. Lack of skills to use and analyse the data. P10. Lack of user analytics data.
<b>6. Organization of the work of development and product teams</b> In this area, the Product Manager carries out or participates in changes in the company related to the organization of work of product and development teams.	P19. Teams are not Agile, they just follow rules and do not use experimentation and the learning process. P44. Teams are not autonomous and self-organised (difficult to organize, dispersed both responsibility and decision-making). P64. Working in silos (problem with communication, synchronisation between teams). P13. The team's lack of motivation due to a lack of understanding of why they are doing this - no commitment to achieving goals.
<b>7. Product Roadmap and stakeholders management</b> In this area, the Product Manager creates a Product Roadmap, uses it to communicate with stakeholders and synchronize the activities of development teams.	P51. Roadmap focused on features instead of goals and business value. P8. Different expectations about product management communication per stakeholder (high/low level). P16. Lack of synchronisation between product management and marketing and sales units. P82. PM role not clearly defined and communicated in the organisation (what the role is about, what responsibilities and objectives are, decisiveness).
<b>8. Other</b> Other issues not covered in the previous categories.	P43. Lack of trust to the product team (micro-management). P27. Unqualified team members (individuals). P41. High expectations from external partners, which are not possible to be met. P56. Price management is always experimentation burdened with risk.

### 5.2.2. Designing focus groups

To ensure the credibility of the study, it was decided that only experienced software product managers could participate. An acceptance threshold of 5 years of experience as a product manager has been settled.

In order to identify and engage such experienced professionals, an initiative called Product Management Challenge was announced. A webpage with the description of the planned research was prepared <sup>1</sup> and distributed in social media groups where software product managers participate. It was also shared by the authors directly through their LinkedIn profiles.

Furthermore, the initiative was promoted through a community that associates product managers, built by *ProductVision.pl*, which Olga Springer co-founded in 2014. This community associates with more than 5.000 product managers in Poland. Each month the website is visited by around 12k users, willing to learn more about product management from the articles, online courses, training and other initiatives organised within the community. *ProductVision.pl* promoted the Product Management Challenge initiative using the following channels: Facebook fanpage, Facebook community group, newsletter, website traffic, LinkedIn profile.

The initiative was promoted under the idea of building a community of experienced software product managers. As a benefit of joining, participants would have the opportunity to share common experiences during focus group sessions. In order to collect data on the software product management experience, the participants of the Product Management Challenge were asked to fill in a questionnaire. Following questions were asked:

1. How many years of experience do you have in the role of Software Product Manager or another role responsible for software product management?
2. How do you self-evaluate your experience related to problems in each category on a scale of 1 to 10 points?
3. What is your name and LinkedIn profile?

As a result of all the activities, 69 software product managers signed up for the initiative. Based on the candidates' experience, a research sample was built. In addition to the 5-year experience requirement, requirement of at least 8 out of 10 points from the self-assessment of experience in at least one problem category was added. Following the selection process, 8 candidates with too little experience were rejected, as they haven't had enough experience as software product managers or they declared their experience in every problem category below defined threshold. Finally, the sample consisted of 61 experienced software product managers.

Table 5.2 shows the number of accepted participants per problem category, their average self-estimation of experience per problem category, as well as the average number of years of experience as software product managers.

---

<sup>1</sup><https://productvision.pl/product-management-challenge/>

**Table 5.2:** Details of focus group participants per category

Category	1	2	3	4	5	6	7	8
Number of experts invited to join focus group in this category	21	29	19	21	42	33	33	8
Number of focus groups carried out	2	2	2	2	2	2	2	1
Number of experts that participated ultimately	16	16	12	10	16	11	13	5
Avg self-estimation of experience of participants (out of 10 points)	8.31	8.37	8.50	8.50	8.56	8.45	8.92	–
Average experience of participants as product managers (years)	7.75	8.00	7.50	7.11	7.80	7.50	8.33	8.50

The original plan was to conduct 8 focus groups, one for each of the identified problem categories. However, given the number of qualified candidates and the recommended number of focus group participants, it was decided to run the research twice independently for all categories except Category 8 (other problems). Therefore, a total of 15 separate focus group interviews were conducted.

### 5.2.3. Conducting the focus group sessions

Altogether, 15 focus groups were organised between December 2020 and March 2021. The average number of participants per focus group interview was 6 experts, which is consistent with the guidelines [117]. Detailed information on participation is presented in Table 5.3.

**Table 5.3:** Execution of focus group study

Category	Number of problems	Meeting duration	Group A: Date	Group A: Number of participants	Group B: Date	Group B: Number of participants
1	3	1.5h	02.12.2020	8	03.12.2020	8
2	2	1.5h	09.12.2020	7	10.12.2020	9
3	3	1.5h	16.12.2020	6	17.12.2020	6
4	3	1.5h	07.01.2021	4	12.01.2021	6
5	4	2h	13.01.2021	8	14.01.2021	8
6	4	2h	20.01.2021	7	21.02.2021	4
7	4	2h	27.01.2021	7	09.02.2021	6
8	4	2h	03.02.2021	5	-	-

Each focus group interview was organized in the same way. The meeting started with an introduction that included welcoming the experts, the presentation of the research goal and method, and the agenda for the meeting. After the introduction, the actual data collection was separated into 2 parts: individual work of experts and teamwork. In the first part, the experts were given problems identified in the category. They were asked to name and describe the solutions they applied in practice to solve them. If experts didn't face the problem, they didn't propose solutions for it. In the second part, the teamwork, each expert elaborated on their solutions, provided context and better explanation and discussed the solutions with other experts. In this part, similar solutions identified independently were merged in the first part upon the mutual agreement of all the authors of the merged solutions, as shown in Table 5.4.

**Table 5.4:** Identified solutions

Category	1		2		3		4		5		6		7		8
Group	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A
All solutions	14	25	19	22	18	26	21	28	36	39	34	17	30	33	21
Merged solutions	13	23	14	19	16	23	20	22	30	38	29	16	29	21	21

During the research, Google Sheets were used as the data collection tool. Shared files made it possible to provide an effective space to prepare descriptions of the solutions to the software product management problems, and merge those solutions that the experts agreed were the same ones. Each description of a solution contained author's identification for easier reference, a short name and detailed steps of the application. Every focus group interview was recorded. The purpose of the recording was to listen to the presentations and discussions in order to verify the descriptions and extend them with some additional details mentioned only verbally. The updated sheets with solutions were further validated with the group of experts who created it.

Identified solutions are reported in the section 3.5.

### 5.3. Validating solutions with Survey

In order to evaluate and select the most comprehensible and effective of the proposed solutions, the questionnaire survey method was applied.

In earlier stages, it was discovered that accessing product managers posed significant challenges. Consequently, a decision was made to streamline the focus and delve deeper into assessing solutions for the five challenges identified as most prevalent in the responsibilities of product managers [19]. As a result, the subsequent problems were chosen for further evaluation:

- **P74.** Determining the true value of the product that the customer needs.
- **P35.** Strategy and priorities are changing frequently.
- **P9.** Technical debt.
- **P64.** Working in silos (problem with communication, synchronisation between teams).
- **P69.** Balancing between reactive and proactive work.

While revising the solutions to these problems, it was noticed that there were still several duplicated solutions to the same problem from two separate focus groups organised for the same category. Therefore, additional merging phase was carried out between the groups.

The results of the merging are shown in Table 5.5.

**Table 5.5:** Second phase of solution merging

Problem	P74		P35		P9		P64		P69	
	I	II	I	II	I	II	I	II	I	II
All solutions	11	9	5	8	9	11	8	13	8	13
Merged solutions within group	8	8	5	8	9	9	8	12	6	12
All solutions from both groups	16		13		18		20		18	
Merged solutions between groups	15		13		18		18		17	

The next stage of the research was to evaluate the usability of the solutions. According to ISO 9241-11 standard, usability is defined as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use [118].

ISO/IEC 25010 provides guidelines for software quality characteristics, including usability. According to ISO/IEC 25010, the attributes of usability are as follows: suitability, understandability, learnability, operability, user error protection, user interface aesthetics, accessibility [119]. These attributes collectively contribute to the usability of software products, helping to ensure that they meet user needs and expectations effectively. The definition of understandability attribute is as follows: the clarity and comprehensibility of the information

presented to users, enabling them to understand the system's functionalities and how to perform tasks.

In order to evaluate the usability of the solutions, two variables were selected: effectiveness and comprehensibility. Similar to understandability, comprehensibility also pertains to the ability to understand something, but it may have a broader connotation. Comprehensibility not only includes the clarity of information but also the ease with which it can be mentally processed or grasped as a whole. It may involve understanding the meaning, significance, or implications of the information or instructions provided. It encompasses factors such as clarity, simplicity, organization, and consistency, which contribute to the ease of understanding [120] [121].

In order to verify the comprehensibility (RQ2) and effectiveness (RQ3) of collected solutions, set of questionnaire surveys were performed. A separate questionnaire was prepared for each of the selected problems. It consisted of a description of the problem and a list of potential solutions. Whereas the description of each solution consisted of its name, and a descriptive solution to be taken to solve the problem. Solutions in questionnaires were presented in non-random order.

For each solution, respondents were asked to rate in a Likert 5-point response format (strongly disagree, disagree, neutral, agree, strongly agree) the accuracy of the following two statements:

1. This solution is comprehensible.
2. This solution solves the problem.

Comprehensibility of the problem solution was defined as the total percentage of "agree" and "strongly agree" responses to the first question, and perceived efficiency as the analogous percentage of responses to the second question. It is important to notice that this definition is based on respondents' opinions, which is typical of qualitative research such as ours.

The experts were engaged through e-mails, and asked them to evaluate the solutions that came out from focus groups. 5 questionnaires to analyse the solutions for each problem separately were sent to 67 experts who signed up for the Product Management Challenge initiative and had a minimum of 5 years of experience in software product management. Each e-mail was sent 1 week after the previous one.

The initial goal was to get at least 17 responses (response rate greater than 30%) for each questionnaire. In order to achieve a satisfying response rate, besides e-mail communication experts were also contacted individually by authors through LinkedIn with requests to complete at least a few surveys they had been invited to.

Table 5.6 presents the number of respondents who took part in evaluating solutions to each problem.

Based on achieved results from the research, as well as discussions during the focus groups, authors defined guidelines for the top frequent problems related to software product management. Guidelines represent high-level insights on how to tackle specific problems.

**Table 5.6:** Participation in the questionnaire survey

Problem	P74	P35	P9	P64	P69
Number of survey responses	31	29	21	21	17
Response rate	46%	43%	31%	31%	25%

In the end, two acceptance thresholds were established for the evaluated solutions, 80% for comprehensibility and 60% for efficiency. Additionally, solutions with negative ratings below 20% (strongly disagree, disagree) were excluded. This resulted in the rejection of 43 out of all 82 solutions. To focus on the high quality of results, higher expectation for the comprehensibility factor was set. The goal was to exclude the results that are hard to read and difficult to interpret.

All literature points out that the implementation of the software product manager roles differs between companies. Different experience and knowledge in software product management roles may be the reason why some software product managers put lower effectiveness on a specific solution. It should be noted that with the defined thresholds, no solution with effectiveness higher than 75% was rejected.

Table 5.7 shows details of the number of accepted solutions for each problem along with the number of solutions rejected based on respective exclusion criteria.

**Table 5.7:** Accepted solutions after the questionnaire survey

Problems	P74	P35	P9	P64	P69
Solutions identified	15	13	18	18	17
Rejected due to both comprehensibility and effectiveness thresholds	3	4	5	5	4
Rejected only due to comprehensibility threshold	2	1	0	1	1
Rejected only due to effectiveness threshold	3	2	3	1	2
Rejected due to negative ratings below 20%	0	1	0	1	0
Accepted solutions	7	5	10	10	10

The list of techniques identified for the most 5 frequent software product management problems is described in section 3.5.

#### **5.4. Creating a list of supporting techniques**

During focus groups, Software Product Managers were sharing their solutions. As part of these solutions, they mentioned also supporting techniques that they use to solve particular problems. Those techniques were caught by my moderator while watching the recordings of the session and gathered as a list of techniques to consider while working on a particular problem.

The list of techniques identified for the most 5 frequent software product management problems is described in table 3.2.

#### **5.5. Summary**

In total, 69 software product managers participated in the Studium 2, aimed at identifying solutions for the most common software product management issues.

Through the execution of focus groups, a compilation of solutions, guidelines, and techniques was formulated, forming the components of the proposed Software Product Management Guide framework.

The use of surveys facilitated the validation of these solutions, assessing their understandability and effectiveness. Solutions failing to meet the necessary thresholds for clarity and effectiveness were excluded.

The SPMG framework, incorporating these novel components, underwent validation in the subsequent study.

## 6. STUDIUM 3 - SPMG FRAMEWORK VALIDATION

The purpose of the studium 3 was to determine if the proposed Software Product Management Guide framework influences software engineering practices in software product development teams.

Software Product Management Guide, as already defined in previous chapters, constitutes a comprehensive set of problems, guidelines, solutions and techniques to evaluate and improve software engineering practices with regard to software product management. The proposed quantification method allows for a numerical representation of how teams perceive problems in their organisations and how they rank potential solutions.

A case study method was chosen for the purpose of the framework validation. A case study in academic research is a term used to describe a research initiative to answer complex research questions, with an empirical approach, using qualitative, quantitative or mixed methods [122]. It was decided to use this method to validate the following hypothesis:

**The proposed Software Product Management Guide framework influences positively software engineering practices in software product development teams.**

### 6.1. Study concept

As Software Product Management Guide framework is a tool aimed to help to organize software development better, it was crucial to bring it into practice and study real-life processes. It was decided to involve teams from the software companies and other organisations and to analyse their real-life processes using SPMG framework.

Each case study main goal was to validate existing challenges related to software product management and search for strategies that software development teams can use to face them. The teams that participate in the study, explore SPMG framework that was designed for that purpose. The framework recommends teams to go through 5 steps: analyse context, diagnose product management state, select strategies, summarise session, create audit report.

The study contains both qualitative and quantitative feedback on the state of product management in software companies. Qualitative methods allow to learn the context and information about the team. They also help to expand the knowledge about existing problems in the team through in-depth questions asked during the session. Open questions are also used to get feedback from the team after they receive the final report. Quantitative methods help teams to validate and prioritize problems that occur in their environment. They also support in selecting the accurate Solutions, Guidelines and Techniques to solve given problems.

Each case study was additionally expanded with evaluation of SPMG framework impact, in order to validate the hypothesis on positive influence.

Table 6.1 presents all methods used to collect data as part of this study.



**Table 6.1:** Data collection methods

Part of the study	Data collection method	Description
Analyse context	Appendix 1 - General Survey, questionnaire	This part is for the session moderator to get familiar with participants based on survey responses.
Diagnose product management state	Appendix 2 - Problems Diagnosis, questionnaire	This part allows the team to identify key problems and score key problems that occur in their organisation and point out if this problem has an impact on their work.
Validation of guidelines, solutions and techniques	Appendix 3 - Solution validation, questionnaire	Participants get familiar with guidelines, solutions and techniques for a given problem, and decide which they could apply in their work.
Framework evaluation	Appendix 3 - Solution validation	Team is questioned if the framework extended their knowledge and if implemented if it could positively impact their software engineering practices.
Open questions during the session	Open questions	Every time before going to the next section or next survey, team should share their thoughts - i.e. about what interesting they found out that would be worth to apply in their project.
After audit feedback	Open questions	Team representative is asked to provide feedback after they have received a summary in the form of an audit.

## 6.2. Case study decisions

In order to receive comparable results from teams it was proposed to unify case studies and carry them out according to the same process. The teams were invited for the 2 hours session. Then the moderator run though 5 steps, using supporting materials (Appendix 1, Appendix 2, Appendix 3).

For the first step (Analyse context) framework proposes to use a questionnaire - Appendix 1 - General Survey to understand the basic information about the company and each participant.

For the second step (Diagnose product management state) framework proposes to ask participants to fill in questionnaire Problem diagnosis - Appendix 2 - in order to identify and prioritise key problems for the team. During step 2 participants are prompted to respond to statements using the Likert scale. These statements included: "I experience this problem" and "This problem has a significant impact on my work." For the statistical analysis of problems, responses that were gathered in the form of labels were assigned with numerical values from -2 to 2. It was done assuming a linear scale and constant distance between different label values.

In order to perform the third step (Select strategies) a questionnaire Solution validation provided in Appendix 3 is proposed. Using this questionnaire participants can individually review and score guidelines, solutions and techniques for the major problems detected in step 2. The number of problems that teams pursue after step 2 is not unified and was decided dynamically using time criterion (every session with a team was planned for 2 hours).

Step 4 (Summarise session) can be performed using data that came from questionnaires used in step 2 and 3. Using the received data moderator is able to quickly recognise key challenges and opportunities that were identified during the session.

The outcome of the study is a software product management audit - a report that the team receives based session results. It is created as part of Step 5 (Prepare a SPMG Audit Report). This report serves as a comprehensive document that the team can utilize for internal reference and decision-making. An example of the SPMG Audit Report can be found as Appendix D. Additionally, it can be shared with company leadership to provide valuable insights into the current state of software product management within the organization.

The report contains a prioritised ranking of problems and potential solutions, guidelines and

techniques to solve one of them. These rankings are created by using responses gathered during the session. A prioritized ranking of problems is compiled and incorporated into the SPM Audit Report. Problem ranking selectively includes only those problems with the highest scores. Each problem in the ranking is accompanied by information detailing the total points allocated based on team responses. The scale of points assigned to each problem statement varies depending on the number of participants in the session. With three participants, the scale ranges from -6 to +6, while with four participants, it extends from -8 to +8.

To prevent overwhelming the team and stakeholders who will receive the SPMG Audit Report with an exhaustive list of problems, a cut-off criterion was introduced for the ranking. While the objectification and optimization of this cut-off threshold may need further exploration, for this research, two criteria were employed.

The first criterion for the prioritized ranking involved including only problems with a positive number of points for both statements. However, there was a singular exception: in Team C, a problem with negative points for significance was incorporated into the ranking. This decision was made after the moderator engaged in a discussion with the team, and together, they unanimously acknowledged the occurrence and significance of this particular problem. The current configuration of the SPMG framework permits threshold customization and allows for exceptions to be made. If, during the session, the moderator identifies a problem as more significant or if it is unanimously acknowledged to occur by everyone in the team, then that problem should be included in the final report.

The second cut-off criterion pertained to the scores assigned to each problem statement. Following the session, the scores for each problem were examined, and only those problems with the highest scores for each team were included in the ranking. This decision was made subjectively by the moderator.

The threshold for inclusion varied between teams due to differences in the number of participants and the number of problems that met the first objective criteria described above. In future research, exploring alternative approaches to designing an objective threshold could be beneficial. Various methods exist for defining and validating an optimal formula. One option is to collaboratively determine with the team during the session how many problems they can realistically address and prioritize within a defined period. Alternatively, presenting all problems with positive scores in the report could provide comprehensive insight. Additionally, leveraging problem weights from previous research, such as a significance factor, may offer valuable insights. This approach entails including only those problems deemed to have a high impact according to experienced software product managers involved in the initial study.

Similar principles that were used to form the ranking of problems for team, were used to create solution rankings. In this study, solutions were validated by soliciting responses to the following statements for each solution: "I can apply this solution in my project," "This solution can solve my problem," and "This solution is comprehensible." Similar to the approach for problems, the same cut-off criterion of a positive number of points was applied. Only solutions with the highest scores for applicability and solvability were included in the ranking. Solution comprehensiveness was not considered a factor that should impact the cut-off threshold. This decision was made because teams have the opportunity to deepen their understanding of solutions if they find them valuable. However, the objectification and determination of an optimal formula for the cut-off threshold for solutions should also be further researched.

The validation of guidelines involved asking the question: "Do you think you could apply that guideline in your project?". Participants were given the following response options: "Yes", "Yes, but", "No" and "No, but". For inclusion in the SPMG Audit Report, only those guidelines with a

positive fraction – indicating a predominance of positive answers (i.e., "Yes" or "Yes, but") – were included. This approach ensured that only guidelines deemed feasible by participants were incorporated into the report.

The same criterion was applied to Techniques as well. The feasibility of utilizing each Technique was assessed with the question: "Do you think you could use this technique in your project?" Participants were provided with response options, including "We already use it", "Yes, we could use it", "No, we can't use it", "I don't know this technique," and "I know the technique but I need more information to decide if we can use it". In the SPMG Audit Report, only those techniques with a positive fraction – indicating a predominance of positive answers (i.e., "Yes, we could use it") – were included. This approach ensured that only techniques considered viable by participants were incorporated into the report. Furthermore, exploring different approaches for evaluating technique feasibility can be considered for future research.

Framework proposes to use shared spreadsheet to generate the rankings of problems and solutions for the team. The moderator can create a document with the results and findings from session i.e. in shared document and share it back with the team. Example of the report can be found in Appendix D.

Software Product Management Guide framework is a set of steps to follow in order to diagnose the product management state and find strategies to improve software engineering practices in organisation. This section summarised decisions to use the SPMG Guide in practice. These decisions can be repeated or modified by the framework users in the future.

The proposed implementation of the steps to use the framework is independent of its evaluation process.

### **6.3. Framework evaluation activities**

In order to validate the framework, all teams after revising problems, guidelines, solutions and techniques for particular problems, as part of the survey (Appendix 3) were asked to rate the following statements:

Provided knowledge within the Software Product Management Guide for this problem:

- Extends my knowledge about potential strategies to solve this particular problem.
- If implemented, will influence positively the software engineering practices that my team uses.

Responses were gathered using a Likert scale. Additionally, the team representatives were requested to share qualitative feedback via e-mail within few days after the session. The following questions were asked:

- If participation in the research was valuable or not? If yes, why?
- How do you rate the framework and the materials it contains? Do you see its use in practice in the future?
- Who can potentially use this framework?

Approximately one month after the session, the team representative was reached out again and asked if team has implemented any changes based on the insights gained during the session and received report.

## 6.4. Case study process

To use the framework, it is proposed to set up a session with the team and run through several steps. It is recommended to secure a minimum of 2 hours for organising the first session - this time will allow to cover a problem diagnosis exercise and exercises for guidelines, solutions and techniques for two selected problems.

It is important to identify a moderator for the sessions - the best if this person has experience in software product management. Such a person will be able to organise discussions with the team, ask in-depth questions to find out root causes and challenge their thoughts.

The study is targeted at representatives of software product teams. Framework guides to invite at least 3 participants from the team, representing different roles - that brings different perspectives and views into discussions. The maximum number of participants is 5 people to not extend the session duration and to allow everyone to speak up.

For bigger organisations, a set of sessions with teams from different departments can be considered, to understand the patterns across the organisation.

In the next subsections lists of activities that the moderator needs to perform is presented, categorized according to the stages of the study.

### 6.4.1. Preparation stage

- Specify the research environment by defining the characteristics of the target participants and organizations. For example, you can determine the ideal size of companies, the type of products developed, and whether the teams receive software product management (PM) support. The scope of this dissertation focuses specifically on software products but was not narrowed down to other criteria.
- Identify a contact within the software team who can facilitate access for your research. Utilize your existing network or expand it as needed by attending conferences where software teams are known to participate. Networking at such events can provide opportunities to establish new contacts and gain access to relevant software teams for your research.
- Invite a team via e-mail - share the purpose of the online session and how they can benefit from it. Propose at least 3 time slots (2 hours) so the invited team can select the best option.
- For the selected date and time, set up a meeting using an online conference tool, i.e. Zoom.
- Send invites to attendees i.e. through Google Calendar

Event Title: Product Management Challenge

Event Description:

*Link to the meeting: to be customised*

*Product Management Challenge - The goal of the initiative is to validate existing challenges related to software product management and search for strategies that software development teams can use to face them.*

*Agenda for the meeting: Problem diagnosis, Strategies for the most severe problems, Closing Diagnosis and summary.*

*Why is it beneficial for your team to participate in the research?*

*Your team is getting access to the framework that was built because of the involvement of 150+ Software Product Managers in the research so far. The team is going through an*

*exercise that will help them understand what are their biggest challenges (related to software product management). Your team will learn and discuss potential solutions that they can use to face the key challenges.*

- Just before the session, ensure all necessary links are prepared for sharing with the team: General survey link (Appendix 1), Problem Diagnosis survey link (Appendix 2) Solution validation links (Appendix 3).

#### **6.4.2. During the session**

- When it's time to begin the online session, launch the meeting platform. Extend a warm welcome to the team, introduce yourself, and reiterate the purpose and agenda for the meeting
- Distribute the link to the Introduction survey (Appendix 1) among the team members and request them to complete it. Once responses are received, take a moment to quickly scan through them to familiarize yourself with the context provided.
- Share the link to the Problem Diagnosis Survey (Appendix 2) with the team. Kindly request the team to inform you once they have responded to all questions. Additionally, ask the team upfront not to save and send the responses before this part is discussed. This ensures that the discussion around the responses can take place before any conclusions are drawn or actions are taken.
- After everyone has completed the survey, invite each participant to share the top 3 main problems that occur in their team/project. That will help to gain a better understanding of each individual's context and perspective. Ask probing questions to clarify any uncertainties. Following the discussion, request the team to save and send their responses so you can assess the numerical representation of problems and summarize the most severe issues for the team.
- Share the results from the Problem Diagnosis section with participants. Highlight the top 5 problems based on their scores for both occurrence and severity. Engage with the team to collectively decide which problem to pursue further during the session.
- Kindly share the link to the Solution Validation Survey (Appendix 3). The form is divided into four sections: Guidelines, Solutions, Techniques, and SPMG Framework Evaluation. Encourage participants to complete each section sequentially. After completing each section, prompt individuals to share their opinions. Use open-ended questions such as:
  - Can you share which guidelines/solutions/techniques would be applicable in your team?
  - Why do you think some guidelines/solutions/techniques cannot be implemented?
  - Which solution do you find most applicable and beneficial for your team?
- Ask the team to save and send responses.
- Promptly review the saved responses and extract key insights from the session. Share these insights with participants, including the primary problems diagnosed, solutions, guidelines, and techniques with the highest scores, as well as any noteworthy strategies or approaches that the team expressed intentions to implement in their projects.

- Share next steps - the team representative will receive SPMG Audit Report and will be requested to share feedback.
- Say thank you and end the session.

#### **6.4.3. Reporting and Documentation Stage**

- Analyse results from all surveys. Decide on cut-off thresholds for rankings and prepare necessary data that have to be added to the SPMG Audit Report (ranking of problems, ranking of solutions, recommended guidelines and techniques).
- Following the session, compile a comprehensive summary of the findings based on participants' responses and discussions. Include all relevant data in the SPMG Audit Report and share it with the team. While the moderator may also offer recommendations based on their experience, it is crucial to clearly distinguish these from the results provided by the SPMG framework. This ensures transparency and clarity in the report's presentation of both empirical findings and expert insights.

#### **6.4.4. Gathering feedback**

- Kindly request feedback from a team representative via email after they have received the SPMG Audit Report summary. Consider using the following open-ended questions:
  - If participation in the research was valuable or not? If yes, why?
  - How do you rate the framework and the materials it contains? Do you see its use in practice in the future?
  - Who can potentially use this framework?
- Additionally approximately one month after the session, reach out to the team representative again and ask if team has implemented any changes based on the insights gained during the session and received SPMG Audit Report.

This section outlined the research protocol, which serves as a guide for replicating the study with other teams.

### **6.5. Participants of case study**

This study was named: Product Management Challenge - Part II and described as an initiative to verify the influence of the new SPMG framework on software engineering practices. Initiative with such value proposition was spread in the product communities, with the goal to recruit teams willing to participate in the study. Eventually for 4 teams signed up and join case study research:

- Team A - Research team from the Gdańsk University of Technology, which develops educational applications that support behavioural therapy for children with autism.
- Team B - One of Polish Online bookstore companies. The people who participated in the meeting worked in different business units and represented different teams.
- Team C - Group of professionals responsible for creating and maintaining software solutions in an airline company and one of the largest airlines in the world.

- Team D - Team developing analytics component in a cloud-based recruitment software company that provides innovative solutions to streamline the hiring process for organizations worldwide.

The Table 6.2 presents the key characteristics of recruited teams.

**Table 6.2:** Teams participated in the research

Characteristics	Team A	Team B	Team C	Team D
Basic info about the team	Research team from Gdańsk University of Technology	One of the polish Online bookstore companies	Group of professionals responsible for creating and maintaining software solutions in airline company	Team who is developing analytics component in cloud-based recruitment software company
Size of the company/organization	1-10	50-250	250+	50-250
Number of participants	3	7 (2 groups)	4	3
Number of members in the team	3-8	5-35	7	6
Research participants	Scrum Master, UX Designer, Project Manager	Main Product Owner, E-commerce Manager, Technical Leader, Product Owner, Customer Service Manager, E-commerce Manager)	Software Developer, Senior Aeronautical Trainer, QA Engineer, Product Owner	Team Leader/Product Manager, Quality Engineer, Product Designer

## 6.6. Case study: Team A

The first team to participate in the research was a team from Gdańsk University of Technology, which developed educational applications that support behavioural therapy for children with autism. The meeting with that team took place on November 3, 2022, and it lasted for 2 hours.

### 6.6.1. Step 1: Analyse context - general survey to get familiar with the team, their project and participants' roles

The initial task assigned to the team was to complete the survey provided in Appendix 1. The results are provided below:

Group characteristics:

1. Size of the company/organization: 1-10
2. Number of participants: 3
3. Number of members in the team: 3-8 (depending on the year)
4. Research participants: Scrum Master, UX Designer, Project Manager
5. Team knowledge about problems related to software product management: Little/medium knowledge

6. Does your team use any systematic method to solve software product management challenges? No or not aware of it
7. Software product management problems that the team was aware of: lack of funds, synchronisation of development of different applications in the project, the applications are developed as masters/engineers thesis, inherited technologies; technological inconsistency, a fluctuating team of developers (knowledge leakage), the need to implement joining participants from scratch, demanding recipient (children), non-technical stakeholders(therapists)

### 6.6.2. Step 2: Diagnose product management state - identify key problems

The second step was a request for a team to fill in the problem validation survey (Appendix 2).

In the table 6.3 below problems that the team experience and have an impact on their work are presented.

**Table 6.3:** Problems identified by the Team A

Problem ID	Problem name	More details about the problem	I experience this problem [Scale of points: -6;6]	This problem has a significant impact of my work [Scale of points: -6;6]
P9	Technical debt	Challenge with developing and maintaining the existing code that is slowing down product development and makes it even more difficult to prioritise the Product Roadmap	6 pt	5 pt
P12	Low software quality	Quality challenges impact customers who are using the product	5 pt	4 pt
P64	Working in silos, problems with communication, synchronization between teams	Initiatives run across different departments require more discipline to align teams around common goals, synchronise them and organise communication that will support team rather than slowing them down.	4 pt	4 pt
P1	No company strategy	When there is no strategy, Product managers and teams struggle with prioritisation, don't see the long term picture and can't achieve outcomes. Requirements and expectations from stakeholders change very often.	5 pt	3 pt
P18	Lack of market research, no understanding of business and trends in the industry	Teams don't have knowledge about what is happening on the market, what are the trends to be able to search for opportunities and valid problems to be solved.	4 pt	0 pt
P6	Lack of continuous integration and delivery (impact on 'time to market')	No proper Engineering standard in place that results in more time needed to release the product to customer.	4 pt	1 pt
Continued on next page				



ID	Problem name	More details about the problem	I experience this problem [Scale of points: -6;6]	This problem has a significant impact of my work [Scale of points: -6;6]
P10	Lack of user analytics data	Teams don't track behavior data from web and mobile applications to achieve a holistic view how user interacts with the product and what is their user experience.	4 pt	1 pt
P27	Unqualified team members (individuals)	Teams struggle with competences of individual team members	2 pt	3 pt
P78	Lack of skills to use and analyze the product/user analytics data	Teams don't have enough knowledge on how to work with user analytics data	2 pt	2 pt
P35	Strategy and priorities are changing frequently	When strategy is changing frequently, product managers and teams struggle with prioritisation and can't achieve outcomes, as direction is changing too often.	2 pt	2 pt

Before moving to the next stage, some time was spent to discuss the challenges.

Observations and comments from participants are listed below as citations:

- *Product management is not on a high level in our project, we feel 75% of the problems.*
- *Out of many problems, 4 problems seem to be the key. Thanks to the meeting we know what we should focus on and we have ideas on how to improve it.*
- *Analysis of the problem provokes reflection.*

### **6.6.3. Step 3: Select strategies - guidelines, solutions and techniques for one of the most significant problems that the team identified**

The next part of the session was when the team was presented with guidelines, solutions and techniques for the identified problem. That team A went through 2 problems using a survey - Appendix 3.

For Technical debt problem (P9), Guidelines that the team perceived as the one that could apply to their project:

- Build debt awareness in the organization, show-specific measures and numbers to educate stakeholders on what is the challenge and why it is worth an investment
- Initiate discussions about technical debt with the team (during scrum ceremonies or additional meetings)
- Prioritise initiatives related to decreasing technical debt in the roadmap or plan constant time each sprint to take care of it.

For Technical debt (P9), solutions that the team perceived as applicable to their projects are presented in table 6.4.

**Table 6.4:** Solutions perceived as applicable by Team A for P9

Solution name	I can apply this solution in my project	This solution is comprehensible	This solution can solve the problem
Solution 10: The boy scout rule - clean up	4 pt	4 pt	3 pt
Solution 5: Product Roadmap – analysis and update	4 pt	5 pt	2 pt
Solution 4: Product Owner taking the initiative – a signal to the team that it is worth taking care of the technical debt.	3 pt	5 pt	2 pt
Solution 6: Managing technical debt	2 pt	5 pt	4 pt

Key things that the team learned from the Software Product Management Guide while exploring the Technical debt problem (P9), that they would like to use in their projects:

- *To leave better code than you find it*
- *Reserve time to work on technical debt*
- *Awareness about debt in teams and organisations is important*
- *Analysis and update of product backlog about technical debt*
- *To talk about debts regularly at the meetings*
- *To try to discuss creating bigger tasks addressing the problem instead of mini tasks.*

For Work in silos problem (P64), Guidelines that the team perceived as the ones that could apply to their project:

- Define strategic product goals to which particular teams would contribute.
- Work on transparent and periodic communication to inform about goals, updates, and learnings
- Set up rituals in order to synchronise the teams: product demos, kick-off meetings before new projects start, regular meetings, roadmapping process

For Work in silos problem (P64), solutions that the team perceived as applicable to their projects are presented in table 6.5.

**Table 6.5:** Solutions perceived as applicable by Team A for P64

Solution name	I can apply this solution in my project	This solution is comprehensible	This solution can solve the problem
Solution 3: Strategic goals	4 pt	4 pt	2 pt
Solution 5: Introduce process to synchronize the work of different teams	2 pt	3 pt	1 pt
Solution 8: Working together on the roadmap by all teams involved in product development	2 pt	2 pt	1 pt
Solution 9: Kick-off meeting before project start	2 pt	4 pt	2 pt

Techniques that the team perceived as could be used in their project for P64:

- Kick-off meeting
- Demo meeting

- Multidisciplinary teams
- Periodic communication

Key things that the team learned from Software Product Management Guide while exploring Working in silos (P64), that they would like to use in their projects:

- *Kick-off meeting with stakeholders*
- *Prioritisation of problems*
- *Try synchronised, regular meetings; periodic meetings with all project participants*
- *Transparency to all project members*

#### **6.6.4. Step 4: Summarise session**

In this Step facilitated discussion was handled by the moderator to the to underline key challenges and opportunities. Moderator shared a screen and displayed data in Google Sheets - responses that team provided during the session. Team discussed top problems that should be prioritized for action (P9, P12, P64, P1) and strategies to explore first based on the framework (for P9 - S10, S5, S4, S6; for P64 - S3, S5, S8, S9). Techniques that could be used were also repeated: Kick-off meeting, Demo meeting, Multidisciplinary teams, Periodic communication.

#### **6.6.5. Step 5: Prepare a SPMG Audit Report**

A SPMG Audit Report created after the session for Team A can be found in Appendix D.

#### **6.6.6. Framework evaluation**

After the team went through guidelines, solutions and techniques for their problems, participants were asked to evaluate the Software Product Management Guide Framework. Teams were asked to respond to Likert's scale for two statements:

Provided knowledge within the Software Product Management Guide for this problem:

- Extends my knowledge about potential strategies to solve this particular problem.
- If implemented, will influence positively the software engineering practices that my team uses.

Table 6.6 presents framework evaluation results from 3 members of Team A. During the step 3 this team evaluated strategies for two separate problems, so they also evaluated the framework twice.

**Table 6.6:** Framework evaluation results by Team A

Statement	Team A
Extend my knowledge about potential strategies to solve this particular problem.	3 x Strongly Agree, 3 x Agree
Influence positively software engineering practices that my team uses.	2 x Strongly Agree, 4 x Agree

The team representative was asked asynchronously for feedback. Three questions were asked:

- If participation in the research was valuable or not? If yes, why?
- How do you rate the framework and the materials it contains? Do you see its use in practice in the future?
- Who can potentially use this framework?

For the first question team representative provided the following feedback back:

- *It's great that you sent a summary of the examination - diagnosis, it gives a lot, to have it collected so officially in one place*
- *Meetings and materials deepened my knowledge of product management, systematized my approach to product management, and showed that I don't have to focus on everything at once, or rather choose 2-3 techniques for today, which reduced the burden associated with implementing product management in our project*
- *Product management diagnosis itself in the project - determining where we are - made me feel more confident in this regard*
- *I realised I approached it like a hedgehog in the past, knowing that something had to be done, but I didn't know where to start*
- *Additionally, even just these 3 people allowed for the confrontation of opinions on various problems and initiated a discussion within the development team*

For the second question team representative provided the following feedback back:

- *The framework is very valuable. I believe that virtually any project or team could go through such an audit.*

For the third question team representative provided the following feedback back:

- *Despite the commercial source (in the sense created on the basis of PM from industry), the project turned out to be very valuable to the non-commercial project.*
- *For product managers who want to improve their management practice*
- *For teams without PMs who see the need to improve in scope product management*
- *For different development roles if they want to understand and improve product management practices*
- *For both small and large teams/projects, rather not to scale globally for the enterprise/larger organization. If you want to use it for a larger organization (many teams), this would require an individual diagnosis of the teams first, and then agreeing on some standard of product stewardship practices suitable for commercial and non-profit products*

Additionally approximately one month after the session, the team representative was reached out again and asked if team has implemented any changes based on the insights gained during the session and received SPMG Audit Report. Response from the representative of Team A for that question was as follows:

- *It is suprising that 2 hours session allowed to identify and change how we manage our software as a product.*

- *A common repository has been created in which materials about all applications are available for newly joining team members.*
- *Two files were created - read me first and read me last. Read Me First contain solution "The Boy Scout rule - clean up"to reduce the technological debt problem. Read Me Last contains information about what needs to be done to complete the project, what is the definition of done, and how to better version to maintain alignment in the delivered pieces of code.*
- *An analysis of application inconsistencies was carried out - in the following categories: technology, UX, content.*
- *After advanced discussions with the team, we decided not to develop the product commercially due to the lack of appropriate technical competencies and business opportunities outside the scientific community.*

## **6.7. Case study: Team B**

The second team that participated in the research was a team representing one of the polish online bookstore companies. The people who participated in the meeting worked in different business units and represented different teams. The meetings with that team took place on November 7, 2022 and February 8, 2023. Both meetings lasted for 2 hours.

**Group 1:** Meeting 1: 07.11.2022, 4 participants: Main Product Owner, Customer Service Manager, E-commerce Manager, Product Owner

**Group 2:** Meeting 2: 08.02.2023, 3 participants: Technical Leader, Product Owner, Main Product Owner

### **6.7.1. Step 1: Analyse context - general survey to get familiar with the team, their project and participants' roles**

The initial task assigned to the team was to complete the survey provided in Appendix 1. The results are provided below:

Group characteristics:

- Size of the company/organization: 50-250.
- Number of participants: 7 (2 groups, 2 sessions)
- Research participants: Main Product Owner (participated in 2 sessions), E-commerce Manager, Technical Leader, Product Owner, Customer Service Manager, E-commerce Manager.
- Number of members in the team: 5-35 (depending on role).
- Team knowledge about problems related to software product management: 5 people with little knowledge, 1 person with advanced knowledge.
- Software product management problems that the team was aware of: There is a lot of that comment from one person with the higher experience in software product management - Main Product Owner, definition and execution of Project Management Flow.
- Does your team use any systematic method to solve software product management challenges? 2 x YES, 2 x NO, 2 x I don't know.

### 6.7.2. Step 2: Diagnose product management state - identify key problems

The second step was a request for teams to fill in the problem validation survey (Appendix 2).

In the table 6.7 problems that the Team B (Group 1 and 2) experience and have an impact on their work are presented.

**Table 6.7:** Problems identified by the teams - merged Group 1 and Group 2

Problem ID	Problem name	More details about the problem	I experience this problem [Scale of points: -14;14]	This problem has a significant impact of my work [Scale of points: -14;14]
P9	Technical debt	Challenge with developing and maintaining the existing code that is slowing down product development and makes it even more difficult to prioritise the Product Roadmap	9 pt	8 pt
P16	Lack of synchronization between product management and marketing and sales units	No resources allocated to the team that would support teams in gaining more knowledge to decide on priorities through research	8 pt	7 pt
P69	Balancing between reactive and proactive work	For mature products it is always a challenge to prioritise research, innovation and new value for customer against queue of bugs, escalation, maintenance related work.	7 pt	7 pt
85	Lack of automated testing	No automated testing, resulting in lowered quality and impact on customers who are using the product	7 pt	8 pt
P64	Working in silos, problem with communication, synchronization between teams	Initiatives run across different departments require more discipline to align teams around common goals, synchronise them and organise communication that will support team rather than slowing them down.	5 pt	9 pt
P8	Different expectations about product management communication per stakeholder (high/low level)	Hard to meet all expectations from stakeholders from different teams and with different seniority levels	5 pt	8 pt
P74	Determining the true value of the product that the customer needs	It is challenging to run extensive research and work iteratively with teams to understand the customer needs, scaling opportunities and customer willingness to pay for the product.	4 pt	4 pt
P12	Low software quality	Quality challenges impact customers who are using the product	4 pt	5 pt

In the tables 6.8 and 6.9 problems that Group 1 and Group 2 experience and have an impact on their work are presented.

**Table 6.8:** Problems identified by the Team B - Group 1

Problem ID	Problem name	More details about the problem	I experience this problem [Scale of points: -8;8]	This problem has a significant impact of my work [Scale of points: -8;8]
P16	Lack of synchronization between product management and marketing and sales units	No resources allocated to the team that would support teams in gaining more knowledge to decide on priorities through research	6 pt	5 pt
P9	Technical debt	Challenge with developing and maintaining the existing code that is slowing down product development and makes it even more difficult to prioritise the Product Roadmap	5 pt	5 pt
P69	Balancing between reactive and proactive work.	For mature products it is always a challenge to prioritise research, innovation and new value for customer against queue of bugs, escalation, maintenance related work.	5 pt	5 pt
P85	Lack of automated testing	No automated testing, resulting in lowered quality and impact on customers who are using the product	4 pt	3 pt
P74	Determining the true value of the product that the customer needs	It is challenging to run extensive research and work iteratively with teams to understand the customer needs, scaling opportunities and customer willingness to pay for the product.	3 pt	3 pt
P64	Working in silos, problem with communication, synchronization between teams	Initiatives run across different departments require more discipline to align teams around common goals, synchronise them and organise communication that will support team rather than slowing them down.	2 pt	4 pt
P8	Different expectations about product management communication per stakeholder (high/low level)	Hard to meet all expectations from stakeholders from different teams and with different seniority levels	2 pt	3 pt
P78	Lack of skills to use and analyze the product/user analytics data	Teams don't have enough knowledge on how to work with user analytics data	2 pt	3 pt
P56	Price management is always experimentation burdened with risk	Teams struggle with decisions about product pricing plans	2 pt	3 pt

**Table 6.9: Problems identified by the Team B - Group 2**

Problem ID	Problem name	More details about the problem	I experience this problem [Scale of points: -6;6]	This problem has a significant impact of my work [Scale of points: -6;6]
P9	Technical debt	Challenge with developing and maintaining the existing code that is slowing down product development and makes it even more difficult to prioritise the Product Roadmap	4 pt	3 pt
P35	Strategy and priorities are changing frequently	When strategy is changing frequently, product managers and teams struggle with prioritisation and can't achieve outcomes, as direction is changing too often	3 pt	5 pt
P64	Working in silos, problem with communication, synchronization between teams	Initiatives run across different departments require more discipline to align teams around common goals, synchronise them and organise communication that will support team rather than slowing them down	3 pt	5 pt
P16	Lack of automated testing	No automated testing, resulting in lowered quality and impact on customers who are using the product	3 pt	5 pt
P8	Different expectations about product management communication per stakeholder	Hard to meet all expectations from stakeholders from different teams and with different seniority levels	3 pt	5 pt
P12	Low software quality	Quality challenges impact customers who are using the product	3 pt	4 pt
P1	No company strategy	When there is no strategy, product managers and teams struggle with prioritisation, don't see the long term picture and can't achieve outcomes. Requirements and expectations from stakeholders change very often	2 pt	5 pt
P82	Product manager role not clearly defined and communicated in the organization	Other teams and individuals don't understand what the product manager role is about, what the responsibilities, objectives, decision-making level	2 pt	4 pt
P19	Teams are not Agile, they just follow rules and do not use experimentation and a learning process	Teams work in iterations and follow some rituals, but as they proceed with work, they don't adapt the process and product.	2 pt	4 pt
P13	The team's lack of motivation due to a lack of understanding of why they are doing things—no commitment to achieve goals	Teams don't understand why they are running some initiatives - they don't believe in the purpose	2 pt	4 pt
P27	Unqualified team members (individuals)	Teams struggle with competences of individual team members	2 pt	4 pt
Continued on next page				



ID	Problem name	More details about the problem	I experience this problem [Scale of points: -6;6]	This problem has a significant impact of my work [Scale of points: -6;6]
P69	Balancing between reactive and proactive work	For mature products it is always a challenge to prioritise research, innovation and new value for customers against a queue of bugs, escalation, maintenance-related work	2 pt	2 pt
P16	Lack of synchronization between product management and marketing and sales units	No resources allocated to the team that would support teams in gaining more knowledge to decide on priorities through research	2 pt	2 pt

Before moving to the next stage, some time was spent to discuss the challenges. Observations and comments from participants (Group 1) are listed below as citations:

- *It's hard to learn product management and translate it into the organization you work for.*
- *We have little customer research, but this does not mean that we do not understand the business.*
- *Technological debt - about 1000 days of work, their problem is also things that have been done recently, currently causing problems, creating functionality takes 2x more time.*
- *They have plenty of data, but they do not aggregate it, they do not use it, they are considering implementing a data warehouse.*

Observations and comments from participants (Group 2) are listed below as citations:

- *We're not trying to solve problems and work iteratively, instead we're trying completely new approaches as soon as we learned our first assumption was wrong. We have a chaotic process of decision-making in such situations.*
- *Engineering doesn't know the big picture of where we are going and what we're trying to achieve. It may affect team motivation.*
- *A Few times a year few strategies change in our company (procedures, software development process). The frequency of changes is quite high. Teams don't have time to get used to the new reality. Lack of understanding of the purposes of these changes due to a lack of understanding of the big picture.*
- *Working with stakeholders is still a challenge - we don't always know what the stakeholders expect and what they're trying to achieve.*

### **6.7.3. Step 3: Select strategies - guidelines, solutions and techniques for one of the most significant problems that the team identified**

The next part of the session was when the team was presented with guidelines, solutions and techniques for the identified problem. That team went through 2 problems using a survey (Appendix 3).

For Determining the true value of the product that the customer needs (P74), Guidelines that the team perceived as the one that could be applicable to their project:

- Investigate the users need in-depth, use both qualitative and quantitative research.
- Collect feedback from users about the product,
- Validate ideas by working on prototypes, close to the customers.

For Determining the true value of the product that the customer needs (P74), solutions that the team B (Group A) perceived as applicable to their projects are presented in table 6.10.

**Table 6.10:** Solutions perceived as applicable by Team B (Group A) for P74

Solution name	I can apply this solution in my project	This solution is comprehensible	This solution can solve the problem
Solution 4: Data analysis and in-depth interviews with clients	6 pt	6 pt	6 pt
Solution 3: In-depth interviews with potential/future users of the product	5 pt	7 pt	4 pt
Solution 1: Customer feedback loop	5 pt	5 pt	4 pt
Solution 2: User testing sessions – testing prototypes	5 pt	5 pt	2 pt
Solution 6: Define Jobs to be done	5 pt	4 pt	3 pt

Key things that the team learned from the Software Product Management Guide while exploring Determining the true value of the product that the customer needs problem (P74), that they would like to use in their projects:

- *Keep concentration on customer*
- *Watch how customers behave*
- *Formalise the way feedback is gathered*
- *Build product roadmap that focus on goals and business value*
- *Invest in data analytics*

Techniques that the team perceived as could be used in their project:

- Customer Feedback Loop
- Kano Model
- Value Proposition Canvas

For Strategy and priorities (P35) are changing frequently, Guidelines that the team B (Group B) perceived as the one that could be applicable to their project:

- Reflect the uncertainty of your strategy in the form of a visual roadmap.
- Develop a strategy with key stakeholders – use results from the research as inputs to the strategy.
- Make sure strategy is communicated.

For Strategy and priorities are changing frequently (P35), solutions that the team perceived as applicable to their projects are presented in table 6.11.

Key things that the team learned from the Software Product Management Guide while exploring Strategy and priorities are changing frequently problem (P35), that they would like to use in their projects:

**Table 6.11:** Solutions perceived as applicable by Team B (Group 2) for P35

Solution name	I can apply this solution in my project	This solution is comprehensible	This solution can solve the problem
Solution 2: Yield potential (incremental) / cost of delay + urgency profile	4 pt	3 pt	3 pt
Solution 3: Taking into account the onboarding of new customers, integrations, and custom-tailored deployments when planning	3	3	3
Solution 4: "Now, Next, Later" roadmap	3 pt	3 pt	4 pt
Solution 5: Define strategy	3 pt	1 pt	2 pt

- *Implementation of OKRs, Vision and Strategy*
- *Consider new Product Management Tools*
- *Product Management environment influence is important*
- *Invest more in Stakeholders management*
- *Involve engineering in goals and vision*

Techniques that the team perceived as could be used in their project:

- Workshops with internal stakeholders
- Impact Analysis
- Product Vision
- Strategy
- OKRs

#### **6.7.4. Step 4: Summarise session**

In this Step facilitated discussion was handled by the moderator to the to underline key challenges and opportunities. Moderator shared a screen and displayed data in Google Sheets - responses that team provided during the session.

Group 1 discussed problems that should be prioritized for action (P16, P9, P69, P85, P74) and recommended solutions to explore first based on the framework (for P74 - S4, S3, S1, S2, S6). Techniques that could be used were also repeated: Customer Feedback Loop, Kano Model, Value Proposition Canvas.

Group 2 discussed problems that should be prioritized for action (P9, P35, P64, P8, P12, P1) and recommended solutions to explore first based on the framework (for P35 - S2, S3, S4, S5). Techniques that could be that could be used were also repeated: Workshops with internal stakeholders, Impact Analysis, Product Vision, Strategy, OKRS.

#### **6.7.5. Step 5: Prepare a SPMG Audit Report**

A SPMG Audit Report created after the session for Team B can be found in Appendix E.

### 6.7.6. Framework evaluation

After the team went through guidelines, solutions and techniques for their problems, participants were asked to evaluate the Software Product Management Guide Framework. Teams were asked to provide responses in Likert's scale for two statements:

- Provided knowledge within Software Product Management Guide for this problem extends my knowledge about potential strategies to solve this particular problem.
- Provided knowledge within Software Product Management Guide for this problem If implemented, will influence positively the software engineering practices that my team uses.

Table 6.12 presents framework evaluation results from 7 participants (Team B).

**Table 6.12:** Framework evaluation results by Team B

Statement	Team B
Extends my knowledge about potential strategies to solve this particular problem.	2 x Strongly Agree, 4 x Agree, 1 x Neutral
Influence positively software engineering practices that my team uses.	3 x Strongly Agree, 3 x Agree, 1 x Neutral

The team representative was asked asynchronously for feedback.

Three questions were asked:

1. If participation in the research was valuable or not? If yes, why?
2. How do you rate the framework and the materials it contains? Do you see its use in practice in the future?
3. Who can potentially use this framework?

For the first question team representative provided the following feedback back:

- *When it comes to conducting the meetings, I really liked it. A good approach of collecting needs in advance, processing people through surveys, which motivates them to think about what you are presenting, even for a moment.*
- *The Software Product Management Guide ends with a toolbox to be used for the problems that have been identified, there is agreement on them, so all that remains is to implement them. Many thinking tools (of course smaller than the framework) are built in such a way as to find the cause, although they do not necessarily show the actual mechanisms or tools with which we can make process corrections. As the presenter, you inquire a lot, you ask a lot, it's good to listen. The things I missed were a small amount of information visualization/tools - it seems to make sense because IT is more of a logic-bound audience, but one image in each case expresses more than 1000 words.*

For the second question team representative provided the following feedback back:

- *Application of knowledge would be valuable, but as I communicated at the meeting, all this later translates into the organization's ability to change and the change management process.*

For the third question team representative provided the following feedback back:

- *Who can use it? In my opinion, to a small extent programmers, Scrum Master and TechLead as well. Knowledge is high-level and collectively covers a large area, therefore, to achieve success there must be the willingness of the individual, the appropriate structure/size of the organization, and an effective process of implementing changes.*

Additionally approximately one month after the session, the team representative was reached out again and asked if team has implemented any changes based on the insights gained during the session and received SPMG Audit Report. Response from the representative of Team B for that question was as follows:

- *Thanks to the diagnostics session using SPMG framework we changed the way we work with stakeholders*
- *Thanks to the inspiration I got from the SPMG framework I made training about OKRs and start using it in my work*
- *We implemented some techniques to improve communication, as we learned soft skills are very important for the success*

## **6.8. Case study: Team C**

The third team to participate in the research was a team creating and maintaining software solutions in an airline company and one of the largest airlines in the world.

The meetings with that team took place on November 3, 2022, and it lasted for 2 hours.

### **6.8.1. Step 1: Analyse context - general survey to get familiar with the team, their project and participants' roles**

The initial task assigned to the team was to complete the survey provided in Appendix 1. The results are provided below.

Group characteristics:

- Size of the company/organization: 250+
- Number of participants: 4 (one person with experience less than 3 months)
- Number of members in the team: 7
- Research participants: Product Owner, Software Developer, Quality Engineer, Domain Specialist
- Team knowledge about problems related to software product management: 2 people with no knowledge, 1 a little, 1 medium
- Software product management problems that the team was aware of: "There are missing features in the new app compared to old software, vertical structure, insufficient communication between Product Owners and Head of Product
- Does your team use any systematic method to solve software product management challenges? 2x No, 2x I don't know

### 6.8.2. Step 2: Diagnose product management state - identify key problems

The second step was a request for teams to fill in the problem validation survey (Appendix 2).

In the table 6.13 below problems that the teams experience and have an impact on their work are presented. 4 members of the team participated in the session, however Quality Engineer had only 2 months of experience in that company. It was decided with the team that the results from that person will not be included. Scale of points in table 6.13 reflects that.

**Table 6.13:** Problems identified by the Team C

Problem ID	Problem name	More details about the problem	I experience this problem [Scale of points: -6;6]	This problem has a significant impact of my work [Scale of points: -6;6]
P10	Lack of user analytics data	Teams don't track behavior data from web and mobile applications to achieve a holistic view how user interacts with the product and what is their user experience.	5 pt	2 pt
P16	Lack of synchronization between product management and marketing and sales units	Product, marketing and sales work separately, don't share same goals and are not aligned	2 pt	1 pt
P64	Working in silos, problem with communication, synchronization between teams	Initiatives run accross diffent departments require more discipline to align teams around common goals, synchronise them and organise communication that will support team rather than slowing them down.	1 pt	-1 pt

Before moving to the next stage, some time was spent to discuss the challenges. Observations and comments from participants are listed below as citations:

- *The company has been focused on meeting the requirements of each client for years, it is hard to switch to a completely different way of thinking. The client needs something, then stakeholders ask when we can deliver this. There is no thinking: will it be useful to everyone? Stakeholders come with solutions.*
- *Alignment with other others - It takes them a lot of time to convince other teams that they need to invest in helping them out to finalise some solution.*
- *The team doesn't know anyone from Sales and Marketing. There are no knowledge sharing and connections between Engineering and these units. Teams don't know how they product is sold and positioned on the market.*

### 6.8.3. Step 3: Select strategies - guidelines, solutions and techniques for one of the most significant problems that the team identified

The next part of the session was when the team was presented with guidelines, solutions and techniques for the identified problem. That team went through 1 problem using a survey (Appendix 3).

For Working in silos, the problem with communication, and synchronization between teams (P64), Guidelines that the team perceived as the one that could be applicable to their project:

- Define strategic product goals to which particular teams would contribute
- Set up rituals in order to synchronise the teams: product demos, kick-off meetings before new projects start, regular meetings, roadmapping process.
- Work on transparent and periodic communication to inform about goals, updates, and learnings.

For Working in silos, the problem with communication, and synchronization between teams (P64), solutions that the team perceived as applicable to their projects:

**Table 6.14:** Solutions perceived as applicable by Team C for P64

Solution name	I can apply this solution in my project	This solution is comprehensible	This solution can solve the problem
Solution 3: Strategic goals	5 pt	6 pt	2 pt
Solution 8: Working together on the roadmap by all teams involved in product development	4 pt	5 pt	3 pt
Solution 9: Kick-off meeting before project start	4 pt	5 pt	3 pt

Key things that the team learned from the Software Product Management Guide while exploring Working in silos, the problem with communication, and synchronization between teams problem (P64), that they would like to use in their projects:

- *Maybe worth trying with kick-offs with the stakeholders and stating clearly the expectations from all sides*
- *Keep pressure to organize Kick-off meetings. Nice to have periodic communication but in a longer period like once per quarter information on what is team doing and what technology is used.*

Techniques that the team perceived as could be used in their project:

- Kick-off meeting

#### **6.8.4. Step 4: Summarise session**

In this Step facilitated discussion was handled by the moderator to the to underline key challenges and opportunities. Moderator shared a screen and displayed data in Google Sheets - responses that team provided during the session. Team discussed the top problems that should be prioritized for action (P10, P16, P64) and recommended solutions to explore first based on the framework (for P64 - S3, S8, S9). Techniques that could be used were also repeated: Kick-off meeting.

#### **6.8.5. Step 5: Prepare a SPMG Audit Report**

A SPMG Audit Report created after the session for Team C can be found in Appendix F.

### 6.8.6. Framework evaluation

After the team went through guidelines, solutions and techniques for their problems, participants were asked to evaluate the Software Product Management Guide Framework. Teams were asked to provide responses in Likert's scale for two statements:

- Provided knowledge within Software Product Management Guide for this problem extends my knowledge about potential strategies to solve this particular problem.
- Provided knowledge within Software Product Management Guide for this problem If implemented, will influence positively the software engineering practices that my team uses.

Table 6.15 presents framework evaluation results from 3 members of Team C.

**Table 6.15:** Framework evaluation results by Team C

Statement	Team A
Extends my knowledge about potential strategies to solve this particular problem.	3 x Neutral
Influences positively software engineering practices that my team uses.	2 x Neutral, 1 x Agree

The team representative was asked asynchronously for feedback. Three questions were asked:

1. If participation in the research was valuable or not? If yes, why?
2. How do you rate the framework and the materials it contains? Do you see its use in practice in the future?
3. Who can potentially use this framework?

The team representative provided one general response to these questions:

*As for the meeting itself - I really liked that you approached the topic in such a comprehensive way, the questions touched on many areas of product development. Our organization is aware of the problems we have in various areas, so nothing really surprised me. As for the proposed solutions without going through the rest of the guide, I am not able to fully comment, but as much I saw during the meeting suggests to me that it is more useful for smaller products, at an earlier stage of development, in organizations that do not yet have until the end of the arranged processes. In our country, the vast majority of the proposed solutions either had already been introduced or we tried to introduce them and for some reason, we gave up on them. The only value I see for a large organization is a trigger to consider whether the solutions we have introduced are definitely the best option.*

Additionally approximately one month after the session, the team representative was reached out again and asked if team has implemented any changes based on the insights gained during the session and received a SPMG Audit Report.

Response from the representative of Team C for that question: no feedback was provided.

### 6.9. Case study: Team D

The fourth team to participate in the research was a team who is developing an analytics component in a cloud-based recruitment software company. Their product provides innovative solutions to streamline the hiring process for organizations worldwide.



The meetings with that team took place on March 31, 2023, and it lasted for 2 hours.

### **6.9.1. Step 1: Analyse context - general survey to get familiar with the team, their project and participants' roles**

The initial task assigned to the team was to complete the survey provided in Appendix 1. The results are provided below.

Group characteristics:

- Size of the company/organization: 50-250
- Number of participants: 3
- Number of members in the team: 6
- Research participants: Team Leader/Product Manager; Quality Engineer; Product Designer
- Team knowledge about problems related to software product management: 1 medium knowledge/ 1 aware of / 1 a little knowledge
- Software product management problems that the team was aware of: "low adoption, missing functionalities, specific needs characteristic to every user, little "data skills"of users; Cost of the feature vs. benefits it will bring; Cutting down the scope at the very last minute"
- Does your team use any systematic method to solve software product management challenges? 2x No, 1x I don't know

### **6.9.2. Step 2: Diagnose product management state - identify key problems**

The second step was a request for participants to fill in the problem validation survey (Appendix 2).

In the table 6.16 problems that the team experience and have an impact on their work are presented.

**Table 6.16:** Problems identified by the Team D

Problem ID	Problem name	More details about the problem	I experience this problem [Scale of points: -6;6]	This problem has a significant impact of my work [Scale of points: -6;6]
P74	Determining the true value of the product that the customer needs	Teams don't track behaviour data from web and mobile applications to achieve a holistic view of how the user interacts with the product and what is their user experience.	4 pt	3 pt
P69	Balancing between reactive and proactive work	For mature products it is always a challenge to prioritise research, innovation and new value for customers against a queue of bugs, escalation, maintenance-related work.	4 pt	3 pt
Continued on next page				

ID	Problem name	More details about the problem	I experience this problem [Scale of points: -6;6]	This problem has a significant impact of my work [Scale of points: -6;6]
P48	Lack of user research	Teams don't spend enough time on research which results in no proper understanding of the problem space and customer side	3 pt	3 pt
P25	Limited access to users in order to do research	Not easy to find potential customers or existing users to perform research, get new knowledge or validate solutions	3 pt	3 pt
P9	Technical debt	Challenge with developing and maintaining the existing code that is slowing down product development and making it even more difficult to prioritise the Product Roadmap	2 pt	2 pt
P16	Lack of synchronization between product management and marketing and sales unit	Product, marketing and sales work separately, don't share the same goals and are not aligned.	5 pt	1 pt
P51	Roadmap focused on features instead of goals and business value	Product Roadmap represents the features that will be developed instead of the problems that will be solved or objectives that are going to be met.	3 pt	1 pt
P64	Working in silos, problems with communication, synchronization between teams	Initiatives run across different departments require more discipline to align teams around common goals, synchronise them and organise communication that will support team rather than slowing them down.	2 pt	1 pt

Before moving to the next stage, some time was spent to discuss the challenges.

### **6.9.3. Step 3: Select strategies - guidelines, solutions and techniques for one of the most significant problems that the team identified**

The next part of the session was when the team was presented with guidelines, solutions and techniques for the 2 identified problems. The team went through 2 problems using a survey (Appendix 3).

For Determining the true value of the product that the customer needs (P74), Guidelines that the team perceived as the one that could be applicable to their project:

- Investigate the users needs in-depth, use both qualitative and quantitative research.
- Collect feedback from users about the product.
- Validate ideas by working on prototypes, close to the customers.

For Determining the true value of the product that the customer needs (P74), solutions that the team perceived as applicable to their projects are presented in table 6.17.

**Table 6.17:** Solutions perceived as applicable by Team D for P74

Solution name	I can apply this solution in my project	This solution is comprehensible	This solution can solve the problem
Solution 4: Data analysis and in-depth interviews with clients	6	6	6
Solution 3: In-depth interviews with potential/future users of the product	5	7	4
Solution 1: Customer feedback loop	5	5	4
Solution 2: User testing sessions – testing prototypes	5	5	2
Solution 6: Define Jobs to be done	5	4	3

Key things that the team learned from the Software Product Management Guide while exploring Determining the true value of the product that the customer needs problem, that they would like to use in their projects:

- *How to define the metrics we want to measure, think about long-term strategy, how to better utilize the sources of user feedback*
- *New strategies and the importance of gathering user feedback through multiple sources, practices to use in planning with a longer perspective in mind*
- *The need for centralised user research (make results easily available for all teams)*
- *Longer-term strategy (think 3-5 years ahead) as it's always a good time to start doing this exercise, to gather feedback via surveys / measure satisfaction around JTBD (upon their completion)*

Techniques that the team perceived as could be used in their project:

- Jobs to be done
- Industry Analysis
- Analysis of contact points with customers (i.e. chat, e-mails)
- Value proposition canvas

For Balancing between reactive and proactive work (P69), Guidelines that the team perceived as the one that could be applicable to their project:

- Guarantee the team a minimum share of proactive work at the level of OKR or team goals.
- Spend time to plan the future – create a vision or north star. Setting product development goals that support this vision and searching for innovations.
- Analyse impact and prioritise initiatives or backlog items in order to define these activities that will bring the greatest value to the user (whether they come from a proactive or reactive pool).

For Balancing between reactive and proactive work (P69), solutions that the team perceived as applicable to their projects can be found in table 6.18:

**Table 6.18:** Solutions perceived as applicable by Team D for P69

Solution name	I can apply this solution in my project	This solution is comprehensible	This solution can solve the problem
Solution 9: Implementation of maintenance and customer service management standards	4 pt	4 pt	4 pt
Solution 7: Define goals (OKRs) and product roadmap	3 pt	3 pt	3 pt
Solution 6: North star	3 pt	4 pt	4 pt
Solution 5: Customer advisory board	3	4	3
Solution 4: Introduce product discovery process in the organisation	2	4	2
Solution 3: Continuous prioritization (e.g. by RICE method)	2	2	2

Key things that the team learned from the Software Product Management Guide while exploring Determining the true value of the product that the customer needs problem, that they would like to use in their projects:

- *RICE, OKR, and Moscow analyses - would love to try them out. Value Proposition Canvas - will utilize it. I will also improve our Jobs to be Done.*
- *New solutions and techniques of balancing work, strategies to implement ways of balancing work on a regular basis.*

Techniques that the team perceived as could be used in their project:

- Design Sprint
- Impact Analysis
- Product Goals
- North Star Metric
- OKRs
- Product Vision
- Industry Analysis
- Jobs to be done

#### **6.9.4. Step 4: Summarise session**

In this Step facilitated discussion was handled by the moderator to the to underline key challenges and opportunities. Moderator shared a screen and displayed data in Google Sheets - responses that team provided during the session. Team discussed the top problems that should be prioritized for action (P74, P69, P48, P25, P9) and recommended solutions to explore first based on the framework (for P74 - S4, S3, S1, S2, S6; for P69 - S9, S7, S6, S5, S4, S3). Techniques that could be used were also repeated: Design Sprint, Impact Analysis, Product Goals, North Star Metric, OKRs, Product Vision, Industry Analysis, Jobs to be done.

#### **6.9.5. Step 5: Prepare a SPMG Audit Report**

A SPMG Audit Report created after the session for Team D can be found in Appendix G.

### 6.9.6. Framework evaluation

After the team went through guidelines, solutions and techniques for their problems, participants were asked to evaluate the Software Product Management Guide Framework. Teams were asked to provide responses in Likert's scale for two statements:

- Provided knowledge within Software Product Management Guide for this problem extends my knowledge about potential strategies to solve this particular problem.
- Provided knowledge within Software Product Management Guide for this problem If implemented, will influence positively the software engineering practices that my team uses.

Table 6.19 presents framework evaluation results from 3 members of Team D. During the step 3 this team evaluated strategies for two separate problems, so they also evaluated the framework twice.

**Table 6.19:** Framework evaluation results by Team D

Statement	Team D
Extend my knowledge about potential strategies to solve this particular problem.	3 x Strongly Agree, 3 x Agree
Influence positively software engineering practices that my team uses.	2 x Strongly Agree, 4 x Agree

The team representative was requested to provide feedback via e-mail, with three questions posed.

1. If participation in the research was valuable or not? If yes, why?
2. How do you rate the framework and the materials it contains? Do you see its use in practice in the future?
3. Who can potentially use this framework?

For the first question team representative provided the following feedback:

*Yes, it was valuable. First of all, due to the possibility of creating a "benchmark" of our team and conducting an examination of conscience with the team ;) Thanks to this, we know where we stand and what are the possible next steps. It was also nice to exchange insights and talk to someone who knows and has dealt with similar problems because the role of a product manager in IT has only recently become a standard.*

For the second question team representative provided the following feedback:

*OK - this is the first and only framework I've used. It seems to me that there was a lack of context for some of our answers and in general, the meeting could have been a bit longer. This way we could go deeper into some topics. The greatest value of this framework is talking to another experienced PM. As for the materials, if it is about methods/suggestions presenter - they in turn require further research and familiarization on your own.*

Additionally approximately one month after the session, the team representative was reached out again and asked if team has implemented any changes based on the insights gained during the session and received SPMG Audit Report. Response from the representative of Team B for that question was as follows:

- *"We have implemented an improved version of the survey examining overall satisfaction with the part of the product we manage. Thanks to it, we receive clear instructions regarding the*

greatest needs. In addition to ad-hoc analysis, I plan to periodically review the responses and, if necessary, continue to contact selected users.

- We also now have a process for launching micro-surveys, which are intended to collect feedback from users before starting work on a new functionality or after its implementation.
- Finally, we cooperate with the tech-support department, which once a quarter compiles a list of problematic cases with which users come.

## 6.10. Comparative analysis of results between teams

4 teams from 4 different companies participated in the research. Table 6.20 presents final results for all teams. The software product management maturity for each team can be examined by the % of problems that the team experiences and the number of potential strategies to apply. The data shows that team A experience the most challenges and have many potential guidelines, solutions and techniques to implement in their project. Team C, on the contrary, doesn't experience so many challenges and already implemented a lot of solutions that the framework suggests. For Team C, the framework still underlined 3 valid problems that the leadership of the organisation should prioritise.

**Table 6.20:** Teams participated in the research

Characteristics	Team A	Team B	Team C	Team D
Basic info about the team	Research team from Gdańsk University of Technology	One of the polish Online bookstore companies	Group of professionals responsible for creating and maintaining software solutions in airline company	Team who is developing analytics component in cloud-based recruitment software company
Size of the company/organization	1-10	50-250	250+	50-250
Number of participants	3	6 (2 groups)	4	3
Number of members in the team	3-8	5-35	7	6
Research participants	Scrum Master, UX Designer, Project Manager	Main Product Owner, E-commerce Manager, Technical Leader, Product Owner, Customer Service Manager, E-commerce Manager)	Software Developer, Senior Aeronautical Trainer, QA Engineer, Product Owner	Team Leader/Product Manager, Quality Engineer, Product Designer
Usage of any systematic method to solve software product management challenges?	no or not aware of it	2x yes, 2x no, 2x I don't know	no or not aware of	no or not aware of it
Number of problems diagnosed	10/27, 37%	8/27, 30%	3/27, 11%	7/27, 26%
Number of solutions evaluated as applicable	8/20, 40%	9/12, 75%	3/10, 30%	11/17, 65%
Number of techniques chosen	4/6, 66%	8/23, 34%	1/6, 16%	12/28, 42%,

Table 6.21 displays the occurrence of each problem within the respective teams, alongside the Frequency and Severity measures from Studium 1 [19]. This data confirms that the problems

identified in Studium 2 also exist and occur within the teams involved in Studium 3. Notably, only four problems—P26, P44, P41, and P3—were not identified in any of the teams during the case studies. This could likely be attributed to the limited sample size.

**Table 6.21:** Problems identified by the teams

Problem ID	Problem name	Team A	Team B	Team C	Team D	Frequency [%]	Severity [%]
P74	Determining the true value of the product that the customer needs		Yes		Yes	72.7	69.3
P35	Strategy and priorities are changing frequently	Yes	Yes			71.6	70.5
P9	Technical debt	Yes	Yes		Yes	70.5	64.8
P64	Working in silos, problem with communication, synchronization between teams	Yes	Yes	Yes	Yes	69.3	65.9
P69	Balancing between reactive and proactive work		Yes		Yes	69.3	63.6
P26	Lack of support for any research (no resources allocated to the team)					65.9	63.6
P85	Lack of automated testing		Yes			65.9	62.5
P82	Product manager role not clearly defined and communicated in the organization		Yes			63.6	58.0
P48	Lack of user research				Yes	61.4	64.8
P51	Roadmap focused on features instead of goals and business value				Yes	61.4	59.1
P18	Lack of market research, no understanding of business and trends in the industry	Yes				55.7	55.7
P8	Different expectations of individual stakeholders regarding product management communication (high/low level)		Yes			55.7	52.3
P6	Lack of continuous integration and delivery (impact on 'time to market')	Yes				55.7	48.9
P25	Limited access to users in order to do research				Yes	54.5	59.1
P16	Lack of synchronization between product management and marketing and sales units		Yes	Yes	Yes	54.5	52.3
P1	No company strategy	Yes	Yes			53.4	61.4
Continued on next page							

Problem ID	Problem name	Team A	Team B	Team C	Team D	Frequency [%]	Severity [%]
P44	Teams are not autonomous and self-organized					53.4	58.0
P10	Lack of user analytics data	Yes		Yes		48.9	54.5
P19	Teams are not Agile, they just follow rules and do not use experimentation and a learning process		Yes			47.7	51.1
P13	The team's lack of motivation due to a lack of understanding of why they are doing things – no commitment to achieve goals		Yes			47.7	46.6
P41	High expectations from external partners, which are not possible to be met					47.7	45.5
P3	Wrong data analytics setup					44.3	44.3
P12	Low software quality	Yes	Yes			43.2	45.5
P78	Lack of skills to use and analyze the data	Yes	Yes			43.2	43.2
P43	Lack of trust in the product team (micro-management)					42.0	51.1
P27	Unqualified team members (individuals)	Yes	Yes			42.0	50.0
P56	Price management is always experimentation burdened with risk		Yes			36.4	29.5

Results presented in Tables 6.20 and 6.21 show that SPMG framework might be used not only to diagnose product management statement in one team, but also to compare teams and organisations.

### 6.11. Thesis validation

The thesis of this dissertation was given as follows:

***The proposed Software Product Management Guide framework influences positively software engineering practices in software product development teams.***

In order to verify this thesis, 3 studiums were carried out. Problems were identified through the use of the systematic literature review, interviews and a survey (Studium 1), while the focus group technique was used to identify solutions, guidelines and supportive techniques (Studium 2). Then case study research was performed to evaluate the framework (Studium 3). All together, over 170 software product managers were involved in the research to form and evaluate the proposed framework.

The main achievement of this thesis - namely Software Product Management Guide constitutes a comprehensive set of problems, guidelines, solutions and techniques to evaluate and improve software engineering practices with regard to software product management. The



proposed framework makes it possible for product development teams to diagnose product management state and review available strategies within just 2 hours. The proposed quantification method in this thesis allows for a numerical representation of how teams perceive problems in their organisations and how they rank potential solutions.

The thesis was validated in the following way. All teams after revising problems, guidelines, solutions and techniques for particular problems, as part of the survey (Appendix 3) were asked to rate the following statements:

Provided knowledge within Software Product Management Guide for this problem:

- Extends my knowledge about potential strategies to solve this particular problem.
- If implemented, will influence positively software engineering practices that my team uses

Responses were gathered using a Likert scale. The table 6.22 presents responses for these statements for all teams. Team A, B and D voted that the framework extended their knowledge and if implemented can influence positively their software engineering practices. Team C, which was a very mature team from a product organisation with long history and experience assessed the impact of the framework as lower.

**Table 6.22:** Thesis validation

Question	Team A	Team B	Team C	Team D
Extend knowledge	3 x Strongly Agree, 3 x Agree	2 x Strongly Agree, 4x Agree, 1 x Neutral	3 x Neutral	3 x Strongly Agree, 3 x Agree
Influence software engineering practices	2 x Strongly Agree, 4 x Agree	3 x Strongly Agree, 3 x Agree, 1 x Neutral	1 x Agree, 2 x Neutral	2 x Strongly Agree, 4 x Agree

### **6.11.1. Statistical analysis**

For the purpose of the statistical analysis, responses that were gathered in the form of labels (Likert scale) were assigned with numerical values from 1 to 5. It was done assuming a linear scale and constant distance between different label values. Some research suggests that for such variables one can calculate mean and standard deviation when the scale is linear (same distance between the values) [123] [124]. So for the variables derived from the questionnaire, mean and standard deviation values were provided in Table 6.23.

Table 6.23 presents results for performed tests. Shapiro statistical test was carried out to check the normality of distributions, and (as expected), the distributions were not normal. It was decided to use a non-parametric test for checking statistical significance. All of the tests were calculated using R function. For *Knowledge Extension* and *Positive influence on software engineering practices* variables the scale ranged 1 to 5, with 5 being the best, and 3 being the neutral score. One-sample Wilcoxon signed rank test with continuity correction was used. For both variables, it was assumed that the value should be significantly higher than 3.5. The tests proved it is which confirm that SPMG framework both improve knowledge and influence software engineering practices.

### **6.11.2. Qualitative feedback on the framework**

As part of the research positive qualitative feedback was received. 3 out of 4 teams confirmed, that they applied solutions and techniques that they learnt from SPMG framework.

**Table 6.23:** Results from statistical analysis

Statistical test	Knowledge extension	Positive Impact on software engineering practices
Standard deviation	0.73	0.66
Mean	4	4
Shapiro-Wilk Normality test	W = 0.80426, p-value = 0.0005838	W = 0.79295, p-value = 0.0003889
Wilcoxon Signed Rank test	V = 223, p-value = 0.0006111	V = 229, p-value = 0.0002755

Quotes from the representative of Team A:

- *It is surprising that 2 hours session allowed to identify and change how we manage our software as a product.*
- *A common repository has been created in which materials about all applications are available for newly joining team members.*
- *Two files were created - read me first and read me last. Read Me First contain solution "The Boy Scout rule - clean up" to reduce the technological debt problem. Read Me Last contains information about what needs to be done to complete the project, what is the definition of done, and how to better version to maintain alignment in the delivered pieces of code.*
- *An analysis of application inconsistencies was carried out - in the following categories: technology, UX, content.*
- *After advanced discussions with the team, we decided not to develop the product commercially due to the lack of appropriate technical competencies and business opportunities outside the scientific community.*

Quotes from the representative of Team B:

- *Thanks to the diagnostical session using SPMG framework we changed the way we work with stakeholders*
- *Thanks to the inspiration I got from the SPMG framework I made training about OKRs and start using it in my work*
- *We implemented some techniques to improve communication, as we learned soft skills are very important for the success*

Quotes from the representative of Team D:

- *"We have implemented an improved version of the survey examining overall satisfaction with the part of the product we manage. Thanks to it, we receive clear instructions regarding the greatest needs. In addition to ad-hoc analysis, I plan to periodically review the responses and, if necessary, continue to contact selected users.*
- *We also now have a process for launching micro-surveys, which are intended to collect feedback from users before starting work on a new functionality or after its implementation.*
- *Finally, we cooperate with the tech-support department, which once a quarter compiles a list of problematic cases with which users come.*

To conclude based on the statistical analysis and qualitative feedback, **performed studies confirmed the thesis of this dissertation.** The proposed SPMG framework help product development teams to optimise their software engineering practices.

## 7. SUMMARY

The primary outcome of this research is the development of a new framework known as the "Software Product Management Guide". This framework was meticulously crafted and evaluated for its impact on software engineering practices through three distinct studies.

Problems related to software product management haven't been studied since 2012. This is why the first study explored the problems that affect the software product management process, their perceived frequency and perceived severity. Data for this investigation were gathered through a systematic literature review (which involved the analysis of five primary databases), interviews with ten software product managers from IT companies, and surveys completed by 89 participants. In total, 95 software product management problems were identified, categorized non-exclusively into seven areas. Among these, 27 frequently cited software product management problems were assessed in terms of their perceived frequency and severity.

The second study explored the topic of software product managers' solutions and techniques for dealing with challenges in their work. The study aims to identify and evaluate solutions to the main problems that affect the software product management process and to provide guidelines for dealing with them. For this purpose, a focus group method was adopted and 47 software product managers participated in 15 focus groups. This was followed by a survey to assess the comprehensibility and effectiveness of the solutions identified. The paper proposes a list of 39 solutions to the 5 most common problems in the work of software product managers.

These two studies allowed the creation of a framework: a Software Product Management Guide. The framework can be used to evaluate software product management across teams and help prioritise the most significant challenges and search applicable strategies. The framework contains 27 problems with perceived frequency and severity, guidelines and solutions for the top 5 frequent problems, and techniques that software product managers use while working on the specific problems. Framework also provides a method in steps that explain how to use it in practice.

The third study evaluates and confirms the framework's positive impact on software engineering practices. 4 case studies were performed with teams from academia and industry. The framework allowed to diagnose key challenges and search for potential strategies during 2-hours sessions. Such tool has high potential in current times - when software product management and being time-effective is very important.

This project proves that solutions to product management problems can influence better software engineering practices if adopted by teams. Effective communication and collaboration are essential to software development. Lack of implemented product management practices can lead to delays, misunderstandings, and unsatisfactory software outcomes. By fostering a culture of transparency, establishing clear roles and responsibilities, using collaboration tools, and encouraging customer feedback, teams can improve communication and collaboration, which can lead to better software outcomes.

Ultimately the guidelines, solutions and techniques to product management problems are aimed at improving the overall product outcome, which includes software engineering practices. By addressing some problems such as a lack of open communication about strategy or data-driven decision-making, teams can work more efficiently and effectively to deliver high-quality software.

The main contribution of the dissertation:

- Listing the main problems specific to product management in software development teams
- Evaluation of the problems with severity and frequency criteria that allows to prioritise attention and focus on what is important
- Defining guidelines for the top problems
- Identification of solutions for top problems
- Rating solutions with comprehensibility and effectiveness
- Listing possible techniques to be employed in the problem-solving
- Combination of the above in a comprehensible framework with descriptions
- Proposing a process for the framework used as a diagnostic tool
- Evaluation of the proposed framework with a guide to applicable in real-life settings

The body of knowledge in this field was updated, which addresses many changes in the IT industry in recent years, including agile and software product management frameworks becoming more widely used.

Furthermore, it may be studied if there is any correlation between problems, solutions, company size, stage of the product lifecycle and other variables.

The research may be also conducted with the top management to check what their perspective is and how they solve the problems related to software product management, as it also affects their business. Finally, a comprehensive ranking of solutions for all identified problems could be developed.

## Bibliography

- [1] Samuel Fricker. "Software Product Management". W: sierp. 2012. ISBN: 978-3-642-31370-7. DOI: 10.1007/978-3-642-31371-4\_4.
- [2] Hans-Bernd Kittlaus i Samuel A Fricker. *Software product management*. Springer, 2017.
- [3] Christof Ebert. "The impacts of software product management". W: *Journal of systems and software* 80.6 (2007), s. 850–861.
- [4] liba. *Babok: A Guide to the Business Analysis Body of Knowledge*. t. 3. International Institute of Business Analysis, 2015. ISBN: 9781927584026. URL: <https://books.google.pl/books?id=ogxTrgEACAAJ>.
- [5] Project Management Institute. *PMBOK® Guide*. Project Management Institute, 2021.
- [6] Jeff Sutherland i Ken Schwaber. *The 2020 Scrum Guide*. Accessed on 17.01.2022. 2020. URL: <https://scrumguides.org/scrum-guide.html>.
- [7] Scaled Agile Inc. *White Paper: Achieving Business Agility with SAFe® 5.0*. 2019. URL: <https://www.scaledagile.com/resources/safe-whitepaper/>. **urldate** 09.08.2023.
- [8] Axelos. *Managing Successful Projects with PRINCE2® (6th edn)*. Axelos, 2017.
- [9] Janeth Lopez i in. "Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review". W: kw. 2016, s. 141–148. DOI: 10.1109/CONISOFT.2016.30.
- [10] Daniel Méndez Fernández i in. "Naming the Pain in Requirements Engineering: Contemporary Problems, Causes, and Effects in Practice". W: *Empirical Software Engineering* (sierp. 2016). DOI: 10.1007/s10664-016-9451-7.
- [11] Pavel Kagan, Anna Naumova i Yuriy Vilman. "The Problems of project management software implementation in construction corporations". W: *MATEC Web of Conferences* 73 (sty. 2016), s. 07016. DOI: 10.1051/mateconf/20167307016.
- [12] Petteri Uusitalo i in. "Solving design management problems using lean design management: the role of trust". W: *Engineering, Construction and Architectural Management* 26 (sierp. 2019), s. 1387–1405. DOI: 10.1108/ECAM-03-2018-0135.
- [13] Andrey Maglyas, Uolevi Nikula i Kari Smolander. "Lean solutions to software product management problems". W: *IEEE software* 29.5 (2012), s. 40–46.
- [14] Andrey Maglyas, Uolevi Nikula i Kari Smolander. "What are the roles of software product managers? An empirical investigation". W: *Journal of Systems and Software* 86 (grud. 2013), s. 3071–3090. DOI: 10.1016/j.jss.2013.07.045.
- [15] Christof Ebert i Sjaak Brinkkemper. "Software product management – An industry evaluation". W: *Journal of Systems and Software* 95 (2014), s. 10–18. ISSN: 0164-1212. DOI: <https://doi.org/10.1016/j.jss.2013.12.042>. URL: <https://www.sciencedirect.com/science/article/pii/S0164121214000156>.
- [16] Andrey Maglyas i in. "Core Software Product Management Activities". W: *Journal of Advances in Management Research* 14 (lut. 2017), s. 23–45. DOI: 10.1108/JAMR-03-2016-0022.
- [17] Sami Jantunen, Kati Hietaranta i Donald Gause. "Adjusting to Increasing Product Management Problems: Challenges and Improvement Proposals in One Software Company". W: t. 7830. Kw. 2013, s. 386–400. ISBN: 978-3-642-37421-0. DOI: 10.1007/978-3-642-37422-7\_28.
- [18] Willem Bekkers i in. "Evaluating the Software Product Management Maturity Matrix". W: wrz. 2012, s. 51–60. ISBN: 978-1-4673-2783-1. DOI: 10.1109/RE.2012.6345839.
- [19] Olga Springer i Jakub Miler. "A comprehensive overview of software product management challenges". W: *Empirical Software Engineering* 27 (wrz. 2022). DOI: 10.1007/s10664-022-10134-5.

- [20] Olga Springer, Jakub Miler i Michał R. Wróbel. "Strategies for Dealing With Software Product Management Challenges". W: *IEEE Access* 11 (2023), s. 55797–55813. DOI: 10.1109/ACCESS.2023.3282605.
- [21] Philip Kotler i in. *Principles of marketing*. Sty. 2003.
- [22] Willem Bekkers i in. "A Framework for Process Improvement in Software Product Management". W: t. 99. Wrz. 2010, s. 1–12. ISBN: 978-3-642-15665-6. DOI: 10.1007/978-3-642-15666-3\_1.
- [23] Greg Geracie i Steven D Eppinger. *The Guide to the Product Management and Marketing Body of Knowledge*. Product Management Educational Institute (PMEI), 2013.
- [24] Hans-Bernd Kittlaus. "Software Product Management and Agile Software Development: Conflicts and Solutions". W: *Software for People*. Red. Alexander Maedche, Achim Botzenhardt i Ludwig Neer. Management for Professionals. Springer, grud. 2012, s. 83–96. DOI: 10.1007/978-3-642-31371-4. URL: [https://ideas.repec.org/h/spr/mgmchp/978-3-642-31371-4\\_5.html](https://ideas.repec.org/h/spr/mgmchp/978-3-642-31371-4_5.html).
- [25] Eric Ries. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation*. Currency, 2011.
- [26] Yves Pigneur Alexander Osterwalder. *A Handbook for Visionaries, Game Changers, and Challengers*. John Wiley i Sons, 2010.
- [27] Inge van de Weerd, Willem Bekkers i Sjaak Brinkkemper. "Developing a Maturity Matrix for Software Product Management". W: *Software Business*. Red. Pasi Tyrväinen, Slinger Jansen i Michael A. Cusumano. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, s. 76–89. ISBN: 978-3-642-13633-7.
- [28] Inge Van de Weerd i in. "A Reference Framework for Software Product Management". W: (sty. 2006).
- [29] Pragmatic Institute. *The pragmatic Framework*. 2020. URL: <https://www.pragmaticinstitute.com/framework/>. **urldate** 03.08.2023.
- [30] Olga Springer i Jakub Miler. "The Role of a Software Product Manager in Various Business Environments". W: *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*. 2018, s. 985–994.
- [31] Nishant A Parikh. "The Software Product Management Framework is Not the Software Product Manager's Framework a Systematic Literature Review". W: *SSRN* (czer. 2023), s. 576–593. DOI: <http://dx.doi.org/10.2139/ssrn.4450114>.
- [32] Timo Wagenblatt. *Software Product Management: Finding the Right Balance for Your Product Inc.* Sty. 2019. ISBN: 978-3-030-19870-1. DOI: 10.1007/978-3-030-19871-8.
- [33] Christof Ebert. "Managing software products in a global context". W: maj 2018, s. 69–76. ISBN: 978-1-4503-5717-3. DOI: 10.1145/3196369.3196371.
- [34] Christof Ebert. "Software Product Management". W: *Software, IEEE* 31 (maj 2014), s. 21–24. DOI: 10.1109/MS.2014.72.
- [35] User Interviews. *Customer Interviews: The Ultimate Guide to Designing and Conducting a Customer Interview*. 2023. URL: <https://www.userinterviews.com/blog/the-ultimate-guide-to-doing-kickass-customer-interviews>. **urldate** 02.08.2023.
- [36] Christine White. *What's a Competitive Analysis How Do You Conduct One?* 2022. URL: <https://blog.hubspot.com/marketing/competitive-analysis-kit>. **urldate** 02.08.2023.
- [37] Devin Pickell. *How to Create a Customer Feedback Loop That Works*. 2023. URL: <https://www.helpscout.com/blog/customer-feedback-loop/>. **urldate** 02.08.2023.
- [38] Rikke Friis Dam i Teo Yu Siang. *Personas – A Simple Introduction*. 2022. URL: <https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them>. **urldate** 02.08.2023.

- [39] Interaction Design Foundation. *How to Conduct User Observations*. 2021. URL: <https://www.interaction-design.org/literature/article/how-to-conduct-user-observations>. **urldate** 02.08.2023.
- [40] Sam Tardif. *What every product manager needs to know about product analytics*. N/A. URL: <https://www.atlassian.com/agile/product-management/product-analytics>. **urldate** 02.08.2023.
- [41] Tony Ulwick. *What Is Jobs-to-be-Done?* 2017. URL: <https://jobs-to-be-done.com/what-is-jobs-to-be-done-fea59c8e39eb>. **urldate** 02.08.2023.
- [42] hotjar. *What is Net Promoter Score®? Your introduction to NPS*. 2023. URL: <https://www.hotjar.com/net-promoter-score/>. **urldate** 02.08.2023.
- [43] Inc. *Industry Analysis*. N/A. URL: <https://www.inc.com/encyclopedia/industry-analysis.html>. **urldate** 02.08.2023.
- [44] Susan Pfundt. *Touchpoint Analysis: How Can You Improve Customer Contact?* 2021. URL: <https://www.zenloop.com/en/blog/touchpoint-analysis/>. **urldate** 03.08.2023.
- [45] microTOOL. *The Kano Model. A Means of Analyzing Customer Desires*. N/A. URL: <https://www.microtool.de/en/knowledge-base/what-is-the-kano-model/>. **urldate** 03.08.2023.
- [46] Daniel Pereira. *What is the Value Proposition Canvas?* 2023. URL: <https://businessmodelanalyst.com/value-proposition-canvas/>. **urldate** 03.08.2023.
- [47] SendPulse. *MVP*. 2023. URL: <https://sendpulse.com/support/glossary/mvp>. **urldate** 03.08.2023.
- [48] Compass. *HOW TO CONDUCT A STAKEHOLDER WORKSHOP*. N/A. URL: <https://thecompassforsbc.org/how-to-guide/how-conduct-stakeholder-workshop>. **urldate** 03.08.2023.
- [49] Mind Tools Content Team. *Impact Analysis*. N/A. URL: <https://www.mindtools.com/a4t4kh3/impact-analysis>. **urldate** 03.08.2023.
- [50] ProductPlan. *RICE Scoring Model*. N/A. URL: <https://www.productplan.com/glossary/rice-scoring-model/>. **urldate** 03.08.2023.
- [51] airfocus. *Product Vision*. N/A. URL: <https://airfocus.com/glossary/what-is-product-vision/>. **urldate** 03.08.2023.
- [52] Bree Davies. *Product Roadmap Guide: What is it How to Create One*. N/A. URL: <https://www.atlassian.com/agile/product-management/product-roadmaps>. **urldate** 03.08.2023.
- [53] Ryan Panchadsaram. *What is an OKR? Definition and Examples*. N/A. URL: <https://www.whatmatters.com/faqs/okr-meaning-definition-example>. **urldate** 03.08.2023.
- [54] Vignaraj Gadvi. *How to identify your North Star Metric*. 2022. URL: <https://www.mindtheproduct.com/how-to-identify-your-north-star-metric/>. **urldate** 03.08.2023.
- [55] Emeritus. *What is Business Strategy – Definition, Importance and Levels*. 2023. URL: <https://emeritus.org/in/learn/what-is-business-strategy/>. **urldate** 03.08.2023.
- [56] Instagantt. *Everything you need to know about the Kick-off Meeting*. N/A. URL: <https://instagantt.com/project-management/what-is-a-kick-off-meeting>. **urldate** 03.08.2023.
- [57] Aha! *What are product goals and initiatives?* 2021. URL: <https://www.aha.io/roadmapping/guide/product-strategy/what-are-product-goals-and-initiatives>. **urldate** 03.08.2023.

- [58] Software Planet Group. *Demo Meetings: The Greatest Thing Since Sliced Bread*. 2018. URL: <https://softwareplanetgroup.co.uk/demo-meetings-the-greatest-thing-since-sliced-bread/>. **urldate** 03.08.2023.
- [59] Jessica Piikkila. *What is SAFe?* N/A. URL: <https://www.atlassian.com/agile/agile-at-scale/what-is-safe>. **urldate** 03.08.2023.
- [60] Indeed Editorial Team. *What Is a Multidisciplinary Team? (With Benefits and Tips)*. 2022. URL: <https://au.indeed.com/career-advice/career-development/multidisciplinary-team>. **urldate** 03.08.2023.
- [61] Bit.ai Editorial Team. *Periodic Report: What is it and How to Create It?* 2022. URL: <https://blog.bit.ai/periodic-report/>. **urldate** 03.08.2023.
- [62] CoSchedule. *What Is a Workshop? - Ultimate Marketing Dictionary*. N/A. URL: <https://coschedule.com/marketing-terms-definitions/workshop>. **urldate** 03.08.2023.
- [63] Max Freedman. *How to Conduct a Market Analysis for Your Business*. 2023. URL: <https://www.businessnewsdaily.com/15751-conduct-market-analysis.html>. **urldate** 03.08.2023.
- [64] Interaction Design Foundation. *Design Sprints*. N/A. URL: <https://www.interaction-design.org/literature/topics/design-sprints>. **urldate** 03.08.2023.
- [65] Tim Vire. *What is a Customer Advisory Board? (Overview, Roles, and Responsibilities)*. 2023. URL: <https://www.onboardmeetings.com/blog/customer-advisory-board/>. **urldate** 03.08.2023.
- [66] Janna Bastow. *How to Implement a Product Discovery Process, Step by Step*. 2023. URL: <https://www.prodpad.com/blog/product-discovery-process/>. **urldate** 03.08.2023.
- [67] airfocus. *MoSCoW Prioritization*. N/A. URL: <https://airfocus.com/glossary/what-is-moscow-prioritization/>. **urldate** 03.08.2023.
- [68] Barbara Kitchenham i Stuart Charters. "Guidelines for performing Systematic Literature Reviews in Software Engineering". W: 2 (sty. 2007).
- [69] Tore Dyba, Barbara A Kitchenham i Magne Jorgensen. "Evidence-based software engineering for practitioners". W: *IEEE software* 22.1 (2005), s. 58–65.
- [70] Claes Wohlin. "Guidelines for snowballing in systematic literature studies and a replication in software engineering". W: *ACM International Conference Proceeding Series* (maj 2014). DOI: 10.1145/2601248.2601268.
- [71] Juliana Medeiros i in. "Quality of software requirements specification in agile projects: A cross-case analysis of six companies". W: *Journal of Systems and Software* 142 (kw. 2018). DOI: 10.1016/j.jss.2018.04.064.
- [72] Sana Ben Nasr i in. "Automated Extraction of Product Comparison Matrices From Informal Product Descriptions". W: *Journal of Systems and Software* 124 (list. 2016). DOI: 10.1016/j.jss.2016.11.018.
- [73] George Valença, Carina Alves i Slinger Jansen. "Strategies for Managing Power Relationships in Software Ecosystems". W: *Journal of Systems and Software* 144 (lip. 2018). DOI: 10.1016/j.jss.2018.07.036.
- [74] Stefan Wiesner, Sara Nilsson i Klaus-Dieter Thoben. "Integrating Requirements Engineering for Different Domains in System Development – Lessons Learnt from Industrial SME Cases". W: *Procedia CIRP* 64 (grud. 2017), s. 351–356. DOI: 10.1016/j.procir.2017.03.013.
- [75] Hiva Alahyari, Tony Gorschek i Richard Berntsson Svensson. "An Exploratory Study of Waste in Software Development Organizations using Agile or Lean approaches: A Multiple Case Study at 14 Organizations". W: *Information and Software Technology* 105 (sierp. 2018). DOI: 10.1016/j.infsof.2018.08.006.



- [76] Nirnaya Tripathi i in. "An anatomy of requirements engineering in software startups using multi-vocal literature and case survey". W: *Journal of Systems and Software* 146 (sierp. 2018). DOI: 10.1016/j.jss.2018.08.059.
- [77] Sezin Yaman i in. "Introducing Continuous Experimentation in Large Software-Intensive Product and Service Organizations". W: *Journal of Systems and Software* 2017 (list. 2017), s. 195–211. DOI: 10.1016/j.jss.2017.07.009.
- [78] Supriya Shrivastava i Urvashi Rathod. "A Risk Management Framework for Distributed Agile Projects". W: *Information and Software Technology* 85 (sty. 2017). DOI: 10.1016/j.infsof.2016.12.005.
- [79] Daniel Ståhl, Torvald Mårtensson i Jan Bosch. "The Continuity of Continuous Integration: Correlations and Consequences". W: *Journal of Systems and Software* 127 (lut. 2017). DOI: 10.1016/j.jss.2017.02.003.
- [80] Brian Fitzgerald i Klaas-Jan Stol. "Continuous Software Engineering: A Roadmap and Agenda". W: *Journal of Systems and Software* 25 (lip. 2015). DOI: 10.1016/j.jss.2015.06.063.
- [81] Ahmad Issa i in. "Industrie 4.0 roadmap: Framework for digital transformation based on the concepts of capability maturity and alignment". W: t. 72. Sty. 2018, s. 973–978. DOI: 10.1016/j.procir.2018.03.151.
- [82] Eero Laukkanen, Juha Itkonen i Casper Lassenius. "Problems, Causes and Solutions When Adopting Continuous Delivery—A Systematic Literature Review". W: *Information and Software Technology* 82 (paź. 2016). DOI: 10.1016/j.infsof.2016.10.001.
- [83] Johannes Holvitie i in. *Technical debt and agile software development practices and processes: An industry practitioner survey*. Kw. 2021.
- [84] Jesse Yli-Huumo, Andrey Maglyas i Kari Smolander. "How do software development teams manage technical debt? – An empirical study". W: *Journal of Systems and Software* 120 (maj 2016). DOI: 10.1016/j.jss.2016.05.018.
- [85] Eveliina Lindgren i Jürgen Münch. "Raising the Odds of Success: The Current State of Experimentation in Product Development". W: *Information and Software Technology* 77 (kw. 2016). DOI: 10.1016/j.infsof.2016.04.008.
- [86] Antônio Pitangueira, Rita Suzana Maciel i Márcio Barros. "Software Requirements Selection and Prioritization using SBSE Approaches: A Systematic Review and Mapping of the Literature". W: *Journal of Systems and Software* 103 (paź. 2014). DOI: 10.1016/j.jss.2014.09.038.
- [87] Sami Jantunen i Donald Gause. "Using a Grounded Theory Approach for Exploring Software Product Management Challenges". W: *Journal of Systems and Software* 95 (wrz. 2014). DOI: 10.1016/j.jss.2014.03.050.
- [88] Meghann Drury-Grogan. "Performance on Agile Teams: Relating Iteration Objectives and Critical Decisions to Project Management Success Factors". W: *Information and Software Technology* (maj 2014). DOI: 10.1016/j.infsof.2013.11.003.
- [89] Waldemar Koczkodaj i in. "Consistency-Driven Pairwise Comparisons Approach to Software Product Management and Quality Measurement". W: sty. 2019, s. 292–305. ISBN: 978-3-319-91445-9. DOI: 10.1007/978-3-319-91446-6\_28.
- [90] Chen Lin. "Data driven product management". W: *IEEE Engineering Management Review* 46 (mar. 2018), s. 16–18. DOI: 10.1109/EMR.2018.2810099.
- [91] Ville Heikkilä i in. "Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson". W: *Empirical Software Engineering* in press (grud. 2017). DOI: 10.1007/s10664-016-9491-z.
- [92] Maria Paasivaara i Casper Lassenius. "Scaling Scrum in a Large Globally Distributed Organization: A Case Study". W: sierp. 2016. DOI: 10.1109/ICGSE.2016.34.

- [93] Sigurhanna Kristinsdottir, Marta Larusdottir i Åsa Cajander. "Responsibilities and Challenges of Product Owners at Spotify - An Exploratory Case Study". W: t. 9856. Sierp. 2016, s. 3–16. ISBN: 978-3-319-44901-2. DOI: 10.1007/978-3-319-44902-9\_1.
- [94] Ashley Morris, Rossi Setchi i P.W. Prickett. "Product Change Management and Future Information Architectures". W: kw. 2016. ISBN: 978-3-319-32096-0. DOI: 10.1007/978-3-319-32098-4\_21.
- [95] Frederico Oliveira, Alfredo Goldman i Viviane Santos. "Managing Technical Debt in Software Projects Using Scrum: An Action Research". W: sierp. 2015, s. 50–59. DOI: 10.1109/Agile.2015.7.
- [96] Garm Lucassen i in. "Bridging the Twin Peaks – The Case of the Software Industry". W: czer. 2015. DOI: 10.1109/TwinPeaks.2015.13.
- [97] Jan Bosch. "Architecting to Ensure Requirement Relevance: Keynote TwinPeaks Workshop". W: maj 2015, s. 1–2. DOI: 10.1109/TwinPeaks.2015.8.
- [98] Timo Lehtinen i in. "Why the Development Outcome Does Not Meet the Product Owners' Expectations?" W: t. 212. Maj 2015. ISBN: 978-3-319-18611-5. DOI: 10.1007/978-3-319-18612-2\_8.
- [99] Aleksander Fabijan, Helena Olsson i Jan Bosch. "Customer Feedback and Data Collection Techniques in Software RD: A Literature Review". W: t. 210. Czer. 2015. ISBN: 978-3-319-19592-6. DOI: 10.1007/978-3-319-19593-3\_12.
- [100] Marko Komssi i in. "Roadmapping problems in practice: value creation from the perspective of the customers". W: *Requirements Engineering* 20 (mar. 2013). DOI: 10.1007/s00766-013-0186-3.
- [101] Farnaz Fotrousi, Katayoun Izadyan i Samuel Fricker. "Analytics for Product Planning: In-Depth Interview Study with SaaS Product Managers". W: czer. 2013. DOI: 10.1109/CLOUD.2013.33.
- [102] Sebastian Barney, Aybuke Aurum i Claes Wohlin. "A product management challenge: Creating software product value through requirements selection". W: *Journal of Systems Architecture - Embedded Systems Design* 54 (czer. 2008), s. 576–593. DOI: 10.1016/j.sysarc.2007.12.004.
- [103] Gabriel Steinhardt. *The Product Manager's Toolkit®*. Sty. 2017. ISBN: 978-3-319-49997-0. DOI: 10.1007/978-3-319-49998-7.
- [104] Andrey Maglyas, Uolevi Nikula i Kari Smolander. "Comparison of Software Product Management Practices in SMEs and Large Enterprises". W: t. 114. Czer. 2012, s. 15–26. ISBN: 978-3-642-30745-4. DOI: 10.1007/978-3-642-30746-1\_2.
- [105] Andrey Maglyas, Uolevi Nikula i Kari Smolander. "What do practitioners mean when they talk about product management?" W: wrz. 2012, s. 261–266. ISBN: 978-1-4673-2783-1. DOI: 10.1109/RE.2012.6345812.
- [106] Andrey Maglyas i Samuel Fricker. "The Preliminary Results from the Software Product Management State-of-Practice Survey". W: czer. 2014. ISBN: 978-3-319-08737-5. DOI: 10.1007/978-3-319-08738-2\_21.
- [107] Euiyoung Kim i in. "User-Centered Design Roadmapping: Anchoring Roadmapping in Customer Value Before Technology Selection". W: *IEEE Transactions on Engineering Management* PP (list. 2020), s. 1–18. DOI: 10.1109/TEM.2020.3030172.
- [108] Andrey Saltan, Slinger Jansen i Kari Smolander. "Decision-making in Software Product Management: Identifying Research Directions from Practice". W: grud. 2018.
- [109] S.E. Hove i Bente Anda. "Experiences from Conducting Semi-structured Interviews in Empirical Software Engineering Research". W: t. 2005. Paź. 2005, 10 pp.-. ISBN: 0-7695-2371-4. DOI: 10.1109/METRICS.2005.24.

- [110] Lawrence Palinkas i in. "Purposeful Sampling for Qualitative Data Collection and Analysis in Mixed Method Implementation Research". W: *Administration and policy in mental health* 42 (list. 2013). DOI: 10.1007/s10488-013-0528-y.
- [111] Committee on the Institutional Means for Assessment of Risks to Public Health, National Research Council. *Risk Assessment in the Federal Government: Managing the Process*. National Academies Press, 1983. ISBN: 978-0-309-03329-5.
- [112] Spencer E. Harpe. "How to analyze Likert and other rating scale data". W: *Currents in Pharmacy Teaching and Learning* 7.6 (2015), s. 836–850. ISSN: 1877-1297. DOI: <https://doi.org/10.1016/j.cptl.2015.08.001>. URL: <https://www.sciencedirect.com/science/article/pii/S1877129715200196>.
- [113] Emerson Murphy-Hill i in. "The Design Space of Bug Fixes and How Developers Navigate It". W: *IEEE Transactions on Software Engineering* 41.1 (2015), s. 65–81. DOI: 10.1109/TSE.2014.2357438.
- [114] Andrew Schuster i in. *Management of Risk: Guidance for Practitioners*. Mar. 2007. ISBN: 9780113310388.
- [115] R. Murray-Webster i Great Britain. Office of Government Commerce. *Management of risk: guidance for practitioners*. Best management practice. Stationery Office, 2010. ISBN: 9780113312740. URL: <https://books.google.pl/books?id=BcM-b4DTttcC>.
- [116] Steve Easterbrook i in. "Selecting empirical methods for software engineering research". W: *Guide to advanced empirical software engineering* (2008), s. 285–311.
- [117] Jyrki Kontio, Johanna Bragge i Laura Lehtola. "The Focus Group Method as an Empirical Tool in Software Engineering". W: sty. 2008, s. 93–116. ISBN: 978-1-84800-043-8. DOI: 10.1007/978-1-84800-044-5\_4.
- [118] International Organization for Standardization (ISO). *ISO 9241-11:1998, Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability*. International Organization for Standardization (ISO), 1998. ISBN: 978-0580301079.
- [119] International Organization for Standardization (ISO). *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*. International Organization for Standardization (ISO), 2011. ISBN: 978-92-67-10484-5.
- [120] Cara Tannenbaum, Joseph A. Greer i Debra L. Roter. "Comprehensibility: A review of definitions, operationalizations, and implications for healthcare communication". W: *Patient Education and Counseling* 96.3 (2014), s. 295–301. DOI: 10.1016/j.pec.2014.06.009.
- [121] Michael Holbrook i Janice Singer. "Towards a Measure of Software Comprehensibility". W: *Journal of Systems and Software* 81.9 (2008), s. 1476–1492. DOI: 10.1016/j.jss.2007.10.005.
- [122] Dan Remenyi. *Case Study Research: The Quick Guide Series*. ACPIL, 2013.
- [123] Han-Ching Chen i Nae-Sheng Wang. "The Assignment of Scores Procedure for Ordinal Categorical Data". W: *TheScientificWorldJournal* 2014 (wrz. 2014), s. 304213. DOI: 10.1155/2014/304213.
- [124] Daniel Fernández i in. "Assigning scores for ordered categorical responses". W: *Journal of Applied Statistics* 47.7 (maj 2020), s. 1261–1281. DOI: 10.1080/02664763.2019.167. URL: <https://ideas.repec.org/a/taf/japsta/v47y2020i7p1261-1281.html>.

## LIST OF FIGURES

2.1	Comparison of Product Management vs Software Product Management . . . . .	12
2.2	Examples of software engineering domains . . . . .	12
2.3	Relation between management, software engineering and software product management . . . . .	13
3.1	Software Product Management Guide . . . . .	21
4.1	Evaluation of the software product management problems in the frequency-severity space . . . . .	63
7.1	General survey - part 1 . . . . .	120
7.2	General survey - part 2 . . . . .	121
7.3	General survey - part 3 . . . . .	122
7.4	Problems diagnosis - the beginning of the survey . . . . .	123
7.5	Problems diagnosis - sample: 2 out of 27 problems in the survey . . . . .	124
7.6	Problems diagnosis - end of survey . . . . .	125
7.7	Strategies for a problem - guidelines . . . . .	126
7.8	Strategies for a problem - sample: 1 out of N solutions in the survey . . . . .	127
7.9	Strategies for a problem - sample: 2 out of N solutions in the survey . . . . .	128
7.10	Strategies for a problem - techniques . . . . .	129
7.11	Strategies for a problem - framework evaluation . . . . .	130

## LIST OF TABLES

2.1	Software product management problems and solutions by Ebert [3] . . . . .	18
2.2	Software product management problems and solutions by Maglyas [13] . . . . .	19
3.1	Evaluation of the perceived frequency and perceived severity of the selected software product management problems [19] . . . . .	25
3.2	List of techniques identified in the study [20] . . . . .	41
4.1	Research methods and data collection techniques used to answer the research questions [19] . . . . .	46
4.2	Detailed search criteria of the systematic literature review [19] . . . . .	46
4.3	Number of papers accepted in each step of systematic literature review [19] . . . .	47
4.4	List of articles from systematic literature review [19] . . . . .	48
4.5	Problem areas and topics extracted from the analyzed literature [19] . . . . .	51
4.6	Examples of the problems identified from the literature [19] . . . . .	52
4.7	Characteristics of interviewed software product managers . . . . .	55
4.8	Software product management problems identified in at least 3 of 10 interviews [19].	55
4.9	Software product management problems found in this research mapped onto problems of Maglyas et al. using recognized common themes [19]. . . . .	57
4.10	Distribution of names of respondents' software product management roles . . . . .	60
4.11	Distribution of respondents' roles in IT anytime in their career . . . . .	60
4.12	Distribution of respondents' company sizes . . . . .	61
4.13	Evaluation of the perceived frequency and perceived severity of the selected software product management problems . . . . .	61
5.1	List of categories and assigned problems . . . . .	66
5.2	Details of focus group participants per category . . . . .	68
5.3	Execution of focus group study . . . . .	68
5.4	Identified solutions . . . . .	68
5.5	Second phase of solution merging . . . . .	69
5.6	Participation in the questionnaire survey . . . . .	70
5.7	Accepted solutions after the questionnaire survey . . . . .	71
6.1	Data collection methods . . . . .	73
6.2	Teams participated in the research . . . . .	79
6.3	Problems identified by the Team A . . . . .	80
6.4	Solutions perceived as applicable by Team A for P9 . . . . .	82
6.5	Solutions perceived as applicable by Team A for P64 . . . . .	82
6.6	Framework evaluation results by Team A . . . . .	83

6.7 Problems identified by the teams - merged Group 1 and Group 2 . . . . .	86
6.8 Problems identified by the Team B - Group 1 . . . . .	87
6.9 Problems identified by the Team B - Group 2 . . . . .	88
6.10 Solutions perceived as applicable by Team B (Group A) for P74 . . . . .	90
6.11 Solutions perceived as applicable by Team B (Group 2) for P35 . . . . .	91
6.12 Framework evaluation results by Team B . . . . .	92
6.13 Problems identified by the Team C . . . . .	94
6.14 Solutions perceived as applicable by Team C for P64 . . . . .	95
6.15 Framework evaluation results by Team C . . . . .	96
6.16 Problems identified by the Team D . . . . .	97
6.17 Solutions perceived as applicable by Team D for P74 . . . . .	99
6.18 Solutions perceived as applicable by Team D for P69 . . . . .	100
6.19 Framework evaluation results by Team D . . . . .	101
6.20 Teams participated in the research . . . . .	102
6.21 Problems identified by the teams . . . . .	103
6.22 Thesis validation . . . . .	105
6.23 Results from statistical analysis . . . . .	106

## **LIST OF APPENDIXES**

1. General Survey
2. Problem Diagnosis
3. Solution Validation
4. Audit for Team A
5. Audit for Team B
6. Audit for Team C
7. Audit for Team D

## Appendix 1: General Survey

**General Survey**

---

Please let me know more about your project and team.

---

**Email \***

Valid email

.....

This form is collecting emails. [Change settings](#)

---

**Your name and Surname \***

Short answer text

.....

---

**Your current role in the project \***

Short answer text

.....

---

**Years of experience in the current role \***

Short answer text

.....

---

**Size of the company/organisation that you work for \***

☐ 1-10

☐ 10-50

☐ 50-250

☐ 250+

**Figure 7.1:** General survey - part 1



Number of team members in your project? \*

Short answer text

.....

Do you create/develop software product in your current project? If yes, what is the product and what problem does it solve? \*

Long answer text

.....

What is the name of the role in your organisation/project that is responsible for software product management \*

☐ Software Product Manager

☐ Product Manager

☐ Head of Product

☐ Chief Product Officer

☐ We don't have such role

☐ I don't know if we have such role

☐ Other...

**Figure 7.2:** General survey - part 2

Do you have knowledge about problems related to software product management? \*

☐ None

☐ Aware of

☐ A little knowledge

☐ Medium knowledge

☐ Advanced knowledge

Can you share problems that you are aware of?

Long answer text

.....

Does your team use any systematic method to solve software product management challenges? \*

☐ Yes

☐ No

☐ I don't know

If yes, which one?

Long answer text


.....

**Figure 7.3:** General survey - part 3

## Appendix 2: Problems Diagnosis

### Problems Diagnosis

Please share if you agree or disagree with the following statements about problems related to software product management.

oolga800@gmail.com [Przełącz konto](#) 

\* Wskazuje wymagane pytanie

Adres e-mail \*

Twój adres e-mail

P69: Balancing between reactive and proactive work. When comparing hypotheses with facts, hypotheses lose in value to facts (such as clients' requests, bugs). Managing requirements instead of identifying problems and opportunities, seeking innovation. \*

More details about the problem: For mature products it is always a challenge to prioritise research, innovation and new value for customer against queue of bugs, escalation, maintenance related work.

	strongly disagree	disagree	neutral	agree	strongly agree
I experience this problem in my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This problem has a significant impact on my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 7.4: Problems diagnosis - the beginning of the survey

P13: The team's lack of motivation due to a lack of understanding of why they are doing things—no commitment to achieve goals \*

More details about the problem: Teams don't understand why they are running some initiatives - they don't believe in the purpose

	strongly disagree	disagree	neutral	agree	strongly agree
I experience this problem in my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This problem has a significant impact on my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

P8: Different expectations about product management communication per stakeholder (high/low level) \*

More details about the problem: Hard to meet all expectations from stakeholders from different teams and with different seniority levels

	strongly disagree	disagree	neutral	agree	strongly agree
I experience this problem in my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This problem has a significant impact on my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Figure 7.5:** Problems diagnosis - sample: 2 out of 27 problems in the survey

P35: Strategy and priorities are changing frequently \*

More details about the problem: When strategy is changing frequently, Product Managers and teams struggle with prioritisation and can't achieve outcomes, as direction is changing too often.

	strongly disagree	disagree	neutral	agree	strongly agree
I experience this problem in my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This problem has a significant impact on my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

P19: Teams are not Agile, they just follow rules and do not use experimentation and a learning process \*

More details about the problem: Teams work in iterations and follow some rituals, but as they proceed with the work, they don't increase their knowledge about customer problems/needs, don't value and prioritise decisions that allow them to learn

	strongly disagree	disagree	neutral	agree	strongly agree
I experience this problem in my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This problem has a significant impact on my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Prześlij

Strona 1 z 1

Wyczyść formularz

Figure 7.6: Problems diagnosis - end of survey

## Appendix 3: Solution validation

**Guidelines**

Problem name: Determining the true value of the product that the customer needs.

More details about the problem: It is challenging to run extensive research and work iteratively with teams to understand the customer needs, scaling opportunities and customer willingness to pay for the product.

**Please read identified guidelines for this problem:**

1. Investigate the user's needs in-depth, use both qualitative and quantitative research.
2. Collect feedback from users about the product.
3. Validate ideas by working on prototypes, close to the customers.

"Investigate the user's needs in-depth, use both qualitative and quantitative research." Do you think you could apply that guideline in your project?

Recommended answers: Yes /Yes, but.... / No / No, but....

Twoja odpowiedź

"Collect feedback from users about the product". Do you think you could apply that guideline in your project?

Recommended answers: Yes /Yes, but.... / No / No, but....

Twoja odpowiedź

"Validate ideas by working on prototypes, close to the customers." Do you think you could apply that guideline in your project?

Recommended answers: Yes /Yes, but.... / No / No, but....

Twoja odpowiedź

Wstecz

Dalej

Wyczyść formularz

Figure 7.7: Strategies for a problem - guidelines

## Solutions

Problem name: Determining the true value of the product that the customer needs.

More details about the problem: It is challenging to run extensive research and work iteratively with teams to understand the customer needs, scaling opportunities and customer willingness to pay for the product.

**Please read solutions recommended by experienced product managers for this problem and please share if agree or disagree with the following statements about these solutions.**

### Solution 1: Customer feedback loop

1. Integration of many communication channels with the client – gathering all customer feedback in one tool, e.g. ProductBoard (chats with clients, Intercom, reports from sales meetings). The Slack channel where the Customer Service Team can submit issues/feedback and everyone can share their product ideas.
2. Automate as much as possible. The option to report feedback should be available in the tools other employees use on daily basis. These channels should be integrated with the feedback management tool.
3. Weekly review of new tickets that appeared in the feedback management tool.
4. Classification of reported ideas – in terms of appropriate tagging or connection with another ticket that already exists.
5. Each ticket, when tagged, is additionally marked with priority and complexity estimation.
6. The tool itself builds a hierarchy of the most frequently reported problems and needs

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I can apply this solution in my project	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This solution is comprehensible	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This solution can solve the problem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Figure 7.8:** Strategies for a problem - sample: 1 out of N solutions in the survey

#### Solution 2: User testing sessions – testing prototypes

1. Form a hypothesis and create a business case.
2. Preparation of a workflow for the process with all dev, UX.
3. Preparation of the prototype/wireframes for testing.
4. User testing session with the target group – i.e. a task for test participants to go through a new registration process. Record this session.
5. Research results, identification of pain points, understanding of client's needs.
6. Recommend further steps for the top-level management.

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I can apply this solution in my project	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This solution is comprehensible	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This solution can solve the problem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

#### Solution 3: In-depth interviews with potential/future users of the product

1. Before starting any work, it is the end-users who define the problems and needs – collect information and requirements from them.
2. Verification of information from customers with stakeholders inside the company (what we must, can, should, cannot provide as part of this application).
3. Review of competitors and their products/systems.
4. Verification of assumptions with other end users (i.e. by showing the prototype/wireframes).
5. Preparation of a list of functionalities satisfying the needs of customers and preparation of the MVP scope (selection of those elements that must be included in it)

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I can apply this solution in my project	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This solution is comprehensible	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This solution can solve the problem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Figure 7.9:** Strategies for a problem - sample: 2 out of N solutions in the survey



Do you think you could use this technique in your project?

	We already use it	Yes, We could use it	No, We can't use it	I don't know the technique	I know it technique but i need more information to decide if we could use it
Customer Interviews	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Competition analysis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Customer Feedback Loop	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Product Analytics - implementation and analysis of customer behaviours	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Jobs to be done	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Net promoter score	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Industry Analysis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Analysis of contact points with customers (i.e. chat, e-mails)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kano Model	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Value Proposition Canvas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Minimum Value Product	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Personas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Wstecz

Dalej

Wyczyść formularz

**Figure 7.10:** Strategies for a problem - techniques

Software Product Management Guide - summary

Problem name: Determining the true value of the product that the customer needs.

More details about the problem: It is challenging to run extensive research and work iteratively with teams to understand the customer needs, scaling opportunities and customer willingness to pay for the product.

Please share your general perception about potential applicability and usage of the framework content to solve the problem:

---

Provided knowledge within Software Product Management Guide for this problem:

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Extend my knowledge about potential strategies to solve this particular problem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
If implemented, will influence positively software engineering practices that my team uses	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

---

Name 3 things that you learned and that you are going to use in your projects

Twoja odpowiedź

---

Wstecz

Prześlij

Wyczyść formularz

**Figure 7.11:** Strategies for a problem - framework evaluation

## Appendix D: Audit for Team A

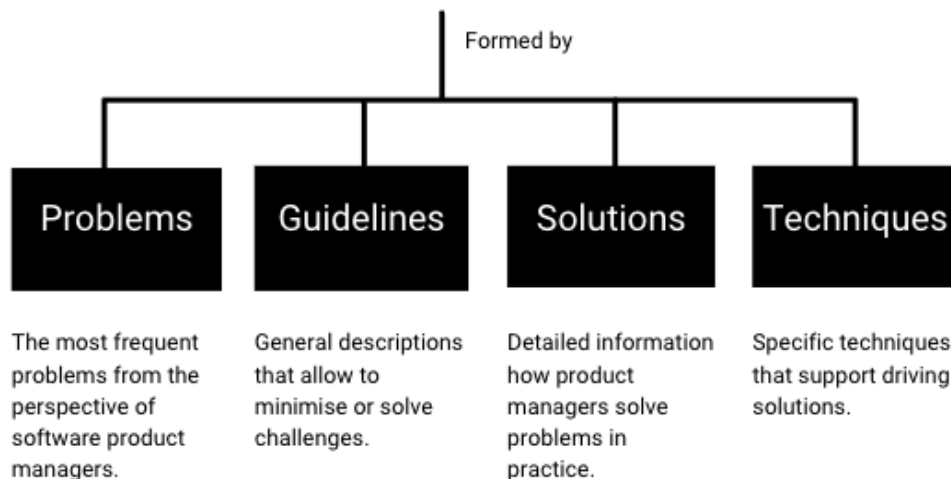
**Software Product Management Guide** is a framework that presents how experienced Software Product Managers solve the problems related to software product management. The framework can be utilised by everyone working on software product development to influence software engineering practices in the organisation.

Software Product Management Guide consists of the following:

1. 27 problems with perceived frequency and severity.
2. Guidelines and solutions for the top 5 frequent problems.
3. Techniques that SPMs use while working on specific problems

\* More than 150 experienced Software Product Managers participated in providing the data to form the framework.

# Software Product Management Guide



## Who is the author of Software Product Management Framework?



Olga Springer is a PhD Student at the Gdańsk University of Technology, Faculty of Electronics and Telecommunications, Department of Software Engineering. She holds a Master's degree in Computer Science. She focuses her research on software product management. She works as Product Manager at Atlassian, previously Head of Product for SentiOne.

# Summary from the Author

Thank you for participating in the research. I am happy that we were able to run the meeting to analyse your software product management challenges.

When I led the set of exercises and the discussion with your teams my general observations were:

- There is no defined product manager role in your team
- Your team experience more than 50% of the problems, which indicates that you have a lot to do to achieve mature product management
- There are 4 key problems, that impact your team the most: Technical debt, Low quality of code, Working in silos, Problems with communication, synchronization between teams, and No company strategy.
- There were voices from the team, that strategy for your product is not clear.

Recommendations based on my own experience:

- Consider creating a business model canvas with the team, to rethink the vision of the product and strategy
- When continuing your project in the University environment, make sure you manage technical debt and quality - build awareness in all teams, and set up rituals to make sure the teams share knowledge and progress around the goals related to their work
- Your team experience this problem: "Lack of market research, no understanding of business and trends in", but says it has no impact on your project. I know your project is pro bono, however, if you work on the technology, it is always good to understand what is happening on the market, and what other segments could potentially benefit from your product. Using data and signals from the market should be very useful as working on the new strategy/product pivot.

You can find detailed results from the research below - separately for each group that participated.

Again, thank you for your time. Wish you a lot of success in the future!

Olga Springer.

Participants:

Scrum Master, UX Designer, Project Manager

Below you can find a list of software product management problems that were recognised by your team:

Problem Name	More details about the problem:	Impact [-6;6]	Significance [-6;6]
P9: Technical debt	Challenge with developing and maintaining the existing code that is slowing down product development and makes it even more difficult to prioritise the Product Roadmap.	6 pt	5 pt
P12: Low software quality	Quality challenges impact customers who are using the product	5 pt	4 pt
P64: Working in silos, problem with communication, synchronization between teams	Initiatives run across different departments require more discipline to align teams around common goals, synchronise them and organise communication that will support team rather than slowing them down.	4 pt	4 pt
P1: No company strategy	When there is no strategy, Product Managers and teams struggle with prioritisation, don't see the long-term picture and can't achieve outcomes. Requirements and expectations from stakeholders change very often.	5 pt	3 pt
P18: Lack of market research, no understanding of business and trends in the industry	Teams don't have knowledge about what is happening on the market, what are the trends to be able to search for opportunities and valid problems to be solved	4 pt	0pt
P6: Lack of continuous integration and delivery (impact on 'time to market')	No proper Engineering standard in place that results in more time needed to release the product to customer.	4 pt	1 pt
P10: Lack of user analytics data	Teams don't track behavior data from web and mobile applications to achieve a holistic view how user interacts with the product and what is their user experience.	4 pt	1 pt
P27: Unqualified	Teams struggle with competences of	2 pt	3 pt

team members (individuals)	individual team members		
P78: Lack of skills to use and analyze the product/user analytics data	Teams don't have enough knowledge on how to work with user analytics data	2 pt	2 pt
P35: Strategy and priorities are changing frequently	When strategy is changing frequently, Product Managers and teams struggle with prioritisation and can't achieve outcomes, as direction is changing too often.	2 pt	2 pt

## Problem: Technical debt

**Guidelines** that the team perceived as the one that could be applicable in their project:

- Build debt awareness in the organization, show-specific measures and numbers to educate stakeholders on what is the challenge and why it is worth an investment
- Initiate discussions about technical debt with the team (during scrum ceremonies or additional meetings)
- Prioritise initiatives related to decreasing technical debt in the roadmap or plan constant time each sprint to take care of it.

**Solutions** that the team perceived as applicable in their projects:

Solution name	I can apply this solution in my project [-6; 6]	This solution is comprehensible [-6; 6]	This solution can solve the problem [-6; 6]
Solution 10: The boy scout rule - clean up	4 pt	4 pt	3 pt
Solution 5: Product Roadmap – analysis and update	4 pt	5 pt	2 pt
Solution 4: Product Owner taking the initiative – a signal to the team that it is worth taking care of the technical debt.	3 pt	5 pt	2 pt
Solution 6: Managing technical debt	2 pt	5 pt	4 pt

Key things that the team learned from Software Product Management Guide, that they would like to use in their projects:

- To leave better code than you find it
- Reserve time to work on technical debt
- Awareness about debt in team and organisation is important
- Analysis and update of product backlog with regard to technical debt
- To talk about debts regularly on the meetings
- To try to discuss creating bigger tasks addressing the problem instead of mini tasks.

## Problem: Work in silos, the problem with communication, synchronisation between teams

**Guidelines** that this team perceived as the one that could be applicable in their project:

- Define strategic product goals to which particular teams would contribute.
- Work on transparent and periodic communication to inform about goals, updates, and learnings
- Set up rituals in order to synchronise the teams: product demos, kick-off meetings before new projects start, regular meetings, roadmapping process.

**Solutions** that the team perceived as applicable in their projects:

Solution name	I can apply this solution in my project [-6; 6]	This solution is comprehensible [-6; 6]	This solution can solve the problem [-6; 6]
Solution 3: Strategic goals	4 pt	4 pt	2 pt
Solution 5: Introduce process to synchronize the work of different	2 pt	3 pt	1 pt
Solution 8: Working together on the roadmap by all teams involved in product development	2 pt	2 pt	1 pt
Solution 9: Kick-off meeting before project star	2 pt	4 pt	2 pt

**Techniques** that the team perceived as could be used in their project:

- Kick-off meeting
- Demo meeting
- Multidisciplinary teams
- Periodic communication

Key things that the team learned from Software Product Management Guide, that they would like to use in their projects:

- Kick-off meeting with stakeholders
- Prioritisation of problems,
- Try synchronised, regular meetings, Periodic meetings with all project participants
- Transparency to all project members,

Observations and comments from research participants:

- Product management is not on a high level in our project, we feel 75 percent of the problems
- Out of many problems, 4 problems seem to be the key. Thanks to the meeting we know what we should focus on and we have ideas on how to improve it.
- Analysis of the problem provokes reflection.



## Appendix E: Audit for Team B

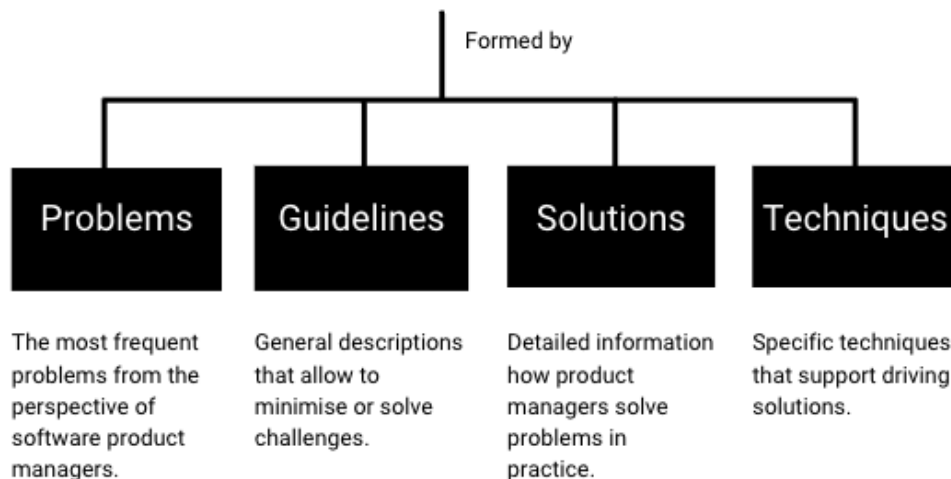
**Software Product Management Guide** is a framework that presents how experienced Software Product Managers solve the problems related to software product management. The framework can be utilised by everyone working on software product development to influence software engineering practices in the organisation.

Software Product Management Guide consists of the following:

1. 27 problems with perceived frequency and severity.
2. Guidelines and solutions for the top 5 frequent problems.
3. Techniques that SPMs use while working on specific problems

\* More than 150 experienced Software Product Managers participated in providing the data to form the framework.

# Software Product Management Guide



## Who is the author of Software Product Management Framework?



Olga Springer is a PhD Student at the Gdańsk University of Technology, Faculty of Electronics and Telecommunications, Department of Software Engineering. She holds a Master's degree in Computer Science. She focuses her research on software product management. She works as Product Manager at Atlassian, previously Head of Product for SentiOne.

# Summary from the Author

Thank you for participating in the research. I am happy that we were able to run 2 separate meetings, to analyse your software product management challenges from the perspective of business roles, as well as from the perspective of engineering.

When I led the set of exercises and the discussion with your teams my general observations were:

- The teams may not be aligned - there is no clear product vision and strategy or goals that they follow (P35, P1, P74)
- There might be a miscommunication between stakeholders and teams - signals: development teams are not aware of the goals behind the changes/features, development teams are not using product analytics and they don't know if someone is actually using them. (P74, P8, P64, P8, P13)
- There may be no clear information what is the difference between Product Manager and Product Owner roles in your company - which may cause confusion and dispersed responsibilities. (P82)
- Technical debt is perceived as one of the most significant problems. (P9)

Recommendations based on my own experience:

1. Consider the implementation of OKRs or another method to execute strategy in your company and make your teams more goal-oriented.
2. For each product in your Portfolio, set up a long-term vision, as well as challenges that are blocking you to achieve it. Design the product strategy together - stakeholders & product managers/product owners & engineering representatives.
3. Increase your product management skills in the organisation. Make sure everyone who hosts PM/PO role has the ability to learn and influence your organisation to try out new approaches and practices. Scrum is one, but increasing the efficiency of software product management processes widely in the organisation - is a different goal.
4. Continue your investment in customer interviews, and ensure your development team has access to customer problems and needs.
5. For Technical Debt, I recommend you review and consider applying [guidelines, solutions and techniques](#) from the Software Product Management Guide. We haven't reviewed and analysed these during our meetings just yet.

You can find detailed results from the research below - separately for each group that participated.

Again, thank you for your time. Wish you a lot of success in the future!

Olga Springer.

# Group 1

\* business perspective - participants:

PM, PO, CSS, E-commerce

Below you can find a list of software product management problems that were recognised by this first team:

Problem name	I experience this problems [-8; 8]	This problem has a significant impact on my work [-8; 8]
P16: Lack of synchronization between product management and marketing and sales units	6 pt	5 pt
P9: Technical debt	5 pt	5 pt
P69: Balancing between reactive and proactive work. When comparing hypotheses with facts, hypotheses lose in value to facts (such as clients requests, bugs)	5 pt	5 pt
P85: Lack of automated testing	4 pt	3 pt
P74: Determining the true value of the product that the customer needs	3 pt	3 pt
P8: Different expectations about product management communication per stakeholder (high/low level)	2 pt	3 pt
P64: Working in silos, problem with communication, synchronization between teams	2 pt	4 pt
P78: Lack of skills to use and analyze the product/user analytics data	2 pt	3 pt
P56: Price management is always experimentation burdened with risk	2 pt	3 pt

## Problem: Determining the true value of the product that the customer needs

**Guidelines** that the team perceived as the one that could be applicable in their project:

- Investigate the users need in-depth, use both qualitative and quantitative research
- Collect feedback from users about the product
- Validate ideas by working on prototypes, close to the customers.

**Solutions** that the team perceived as applicable in their projects:

Solution name	I can apply this solution in my project [-8; 8]	This solution is comprehensible [-8; 8]	This solution can solve the problem [-8; 8]
Solution 4: Data analysis and in-depth interviews with clients	6 pt	6 pt	6 pt
Solution 3: In-depth interviews with potential/future users of the product	5 pt	7 pt	4 pt
Solution 1: Customer feedback loop	5 pt	5 pt	4 pt
Solution 2: User testing sessions – testing prototypes	5 pt	5 pt	2 pt
Solution 6: Define Jobs to be done	5 pt	4 pt	3 pt

**Techniques** that the team perceived as could be used in their project:

- Customer Feedback Loop
- Kano Model
- Value Proposition Canvas

Key things that the team learned from Software Product Management Guide, that they would like to use in their projects:

- Customer concentration
- Watching customers behaviours
- Formalising the way feedback is gathered
- Roadmap focus on goals and business value
- Invest in Data Analytics

## Group 2

\* product and engineering perspective - participants: PM, PO, Team Lead

Below you can find a list of software product management challenges that were recognised by this team:

Problem name	I experience this problem in my work [-6; 6]	This problem has a significant impact on my work [-6; 6]
P9: Technical debt	4 pt	3 pt
P35: Strategy and priorities are changing frequently	3 pt	5 pt
P64: Working in silos, problem with communication, synchronization between teams	3 pt	5 pt
P85: Lack of automated testing	3 pt	5 pt
P8: Different expectations about product management communication per stakeholder	3 pt	5 pt
P12: Low software quality	3 pt	4 pt
P1: No company strategy	2 pt	5 pt
P82: Product Manager role not clearly defined and communicated in the organization (what the role is about, what the responsibilities and objectives are, decisiveness)	2 pt	4 pt
P19: Teams are not Agile, they just follow rules and do not use experimentation and a learning process	2 pt	4 pt
P13: The team's lack of motivation due to a lack of understanding of why they are doing things—no commitment to achieve goals	2 pt	4 pt
P27: Unqualified team members (individuals)	2 pt	4 pt
P69: Balancing between reactive and proactive work. When comparing hypotheses with facts, hypotheses lose in value to facts (such as clients' requests, bugs).	2 pt	2 pt
P16: Lack of synchronization between product management and marketing and sales units	2 pt	2 pt

## Problem: Strategy and priorities are changing frequently

**Guidelines** that this team perceived as the one that could be applicable in their project:

- Create product vision with the most important stakeholders, define the main challenges and business goals.
- Reflect the uncertainty of your strategy in the form of a visual roadmap
- Develop a strategy with key stakeholders – use results from the research as inputs to the strategy.
- Make sure strategy is communicated.

**Solutions** that the team perceived as applicable in their projects:

Solution name	I can apply this solution in my project [-6; 6]	This solution is comprehensible [-6; 6]	This solution can solve the problem [-6; 6]
Solution 2: Yield potential (incremental) / cost of delay + urgency profile	4 pt	3 pt	3 pt
Solution 3: Taking into account the onboarding of new customers, integrations, and custom-tailored deployments when planning	3 pt	3 pt	3 pt
Solution 4: “Now, Next, Later” roadmap	3 pt	3 pt	4 pt
Solution 5: Define strategy	3 pt	1 pt	2 pt

**Techniques** that the team perceived as it could be used in their project:

- Workshops with internal stakeholders
- Impact Analysis
- Product Vision
- Strategy
- OKRs

Key things that the team learned from Software Product Management Guide, that they would like to use in their projects:

- Implementation of OKRs, Vision and Strategy
- Consider Product Management Tools
- Product Management environment influence is important
- Stakeholders management
- Involve engineering in goals and vision

## Merged results from two groups

Below you can find a list of software product management problems that were recognised by your teams (based on results from all 7 participants):

Problem name	I experience this problem [-14; 14]	This problem has a significant impact on my work [-14; 14]
P9: Technical debt	9 pt	8 pt
P16: Lack of synchronization between product management and marketing and sales units	8 pt	7 pt
P69: Balancing between reactive and proactive work.	7 pt	7 pt
P85: Lack of automated testing	7 pt	8 pt
P64: Working in silos, problem with communication, synchronization between teams	5 pt	9 pt
P8: Different expectations about product management communication per stakeholder (high/low level)	5 pt	8 pt
P74: Determining the true value of the product that the customer needs	4 pt	4 pt
P12: Low software quality	4 pt	5 pt

## Appendix F: Audit for Team C

# Research Summary

Thank you for participating in the research.

When I led the set of exercises and the discussion with your team my general observations were:

- The team doesn't have access to user analytics data - this is the key challenge that your team recognise (P10).
- Team feel the effects of working in silos (P16). When they need something from teams that are not contributing to the same goal their requests are undermined, sometimes even rejected.
- Teams don't have visibility into data and knowledge related to marketing and sales (P16). They don't know how the product is sold, what are the key challenges with selling or what are the rejection or churn reasons from the customers.
- There may be no clear information what is the difference between Chief Product Officer, and Senior Product Manager in your company - which may cause confusion and dispersed responsibilities. (P82)
- In general the team is proud of the way they work now - how they set up goals, and how they organise rituals to develop their product. During the problem detection exercise, the team pointed out only 4 problems (out of 27 on the list). The Product Manager assigned to that team recognised 7 problems only, which is a good score.
- On the other side, the team during the session mentioned that there might be other teams in your organisation that are not using similar practices, and these teams may need more attention.

Recommendations based on my own experience:

- The team doesn't have access to user analytics data. Of course, it is not an easy initiative from the compliance perspective, to propose data tracking for enterprise customers, but there are companies that do it successfully. The team claims it is technically possible, but management is pushing for features and they can't find the time to invest in analytics. Consider prioritising this topic in order to increase the effectiveness and maturity of the software product management process in your company.

You can find detailed results from the research below - separately for each group that participated.

Again, thank you for your time. Wish you a lot of success in the future!

Olga Springer.



Participants:

Product Manager, Developer, Senior Aeronautical Trainer

Below you can find a list of software product management problems that were recognised by this team:

Problem name	I experience this problem [-6; 6]	This problem has a significant impact on my work. [-6; 6]
P10: Lack of user analytics data	5 pt	2 pt
P16: Lack of synchronization between product management and marketing and sales units	2 pt	1 pt
P64: Working in silos, the problem with communication, synchronization between teams	1 pt	-1 pt

Problem: Working in silos, the problem with communication, and synchronization between teams

Guideline around which you can improve in your organisation:

Guideline	Feedback
Define strategic product goals to which particular teams would contribute	“in a way it is already happening through OKR process, but might not be communicated clearly enough throughout the organization yet”
Set up rituals in order to synchronise the teams: product demos, kick off meetings before new projects start, regular meetings, roadmapping process.	“for most of the teams it is already in place, but the organization is too large that everyone could attend everything that is interesting for him.  But in my opinion, the biggest problem we have is with the teams that <b>don't have anything in place.</b> ”
Work on transparent and periodic communication to inform about goals, updates, and learnings.	“Development team host Review meetings when they share the outcomes of the work. Maybe, if sales and marketing could share their goals, news, updates as well that could help.”

**Solutions** that the team perceived as applicable in their projects:

Problem name	I can apply this solution in my project [-6; 6]	This solution is comprehensible [-6; 6]	This solution can solve the problem [-6; 6]
Solution 3: Strategic goals	5 pt	6 pt	2 pt
Solution 8: Working together on the roadmap by all teams involved in product development	4 pt	5 pt	3 pt
Solution 9: Kick-off meeting before project start	4 pt	5 pt	3 pt

**A technique** that the team perceived as could be used in their project:

- Kick-off meeting

Key things that the team learned from Software Product Management Guide, that they would like to use in their projects:

- Maybe worth trying with kick-offs with the stakeholders and stating clearly the expectations from all sides
- Keep pressure to organize Kick-off meetings. Nice to have periodic communication but in a longer period like once per quarter information on what is team doing and what technology is used.

## Appendix G: Audit for Team D

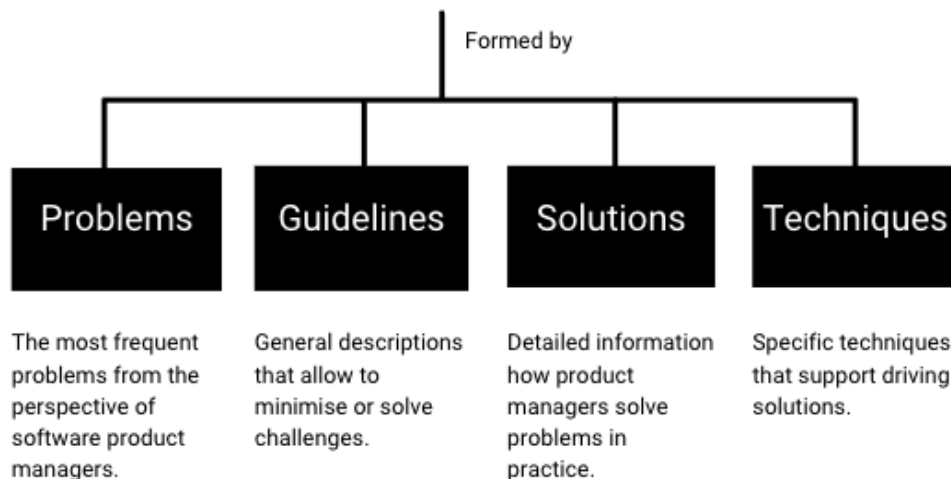
**Software Product Management Guide** is a framework that presents how experienced Software Product Managers solve the problems related to software product management. The framework can be utilised by everyone working on software product development to influence software engineering practices in the organisation.

Software Product Management Guide consists of the following:

1. 27 problems with perceived frequency and severity.
2. Guidelines and solutions for the top 5 frequent problems.
3. Techniques that SPMs use while working on specific problems

\* More than 150 experienced Software Product Managers participated in providing the data to form the framework.

# Software Product Management Guide



## Who is the author of Software Product Management Framework?



Olga Springer is a PhD Student at the Gdańsk University of Technology, Faculty of Electronics and Telecommunications, Department of Software Engineering. She holds a Master's degree in Computer Science. She focuses her research on software product management. She works as Senior Product Manager at Atlassian, previously Head of Product for SentiOne.

# Research Summary

Thank you for participating in the research.

When I led the set of exercises and the discussion with your team my general observations were as follows:

- There is a company and product strategy in the organisation, but your team is working on a specific area, which has lower priorities than others. The team lack a clear vision and long-term strategy for the area that they work on. (P74, P69, P51)
- The team don't have enough opportunities to learn customer needs and behaviours (P74, P48, P51)

Recommendations based on my own experience:

- Consider setting up metrics that show if/how your area contributes to the product strategy. Define North Star Metric for your area to show the progress of your initiatives.
- Run market research and define Long Term Strategy for your area to share in the organisation how the area contributes to the future success of the company.
- Utilise data from all customer touchpoints that you have now - customer escalations, chats, e-mails.
- If your research shows that your area should not be developed further (i.e. because it is not related to key Jobs To be Done for customers that should be optimised), focus on decreasing the maintenance cost of the area and moving people to other initiatives that contribute to other initiatives.

You can find detailed results from the session below.

Again, thank you for your time. Wish you a lot of success in the future!

Olga Springer.

Participants:

Product Manager/Team Lead, QE Engineer, Product Designer

Below you can find a list of software product management problems that were recognised by this team:

Problem name	I experience this problem [-6; 6]	This problem has a significant impact on my work. [-6; 6]
P74: Determining the true value of the product that the customer needs [I experience this problem in my work]	4 pt	3 pt
P69: Balancing between reactive and proactive work. When comparing hypotheses with facts, hypotheses lose in value to facts (such as clients' requests, and bugs).	4 pt	3 pt
P48: Lack of user research	3 pt	3 pt
P9: Technical debt	2 pt	2 pt
P25: Limited access to users in order to do research	3 pt	3 pt
P16: Lack of synchronization between product management and marketing and sales unit	5 pt	1 pt
P51: Roadmap focused on features instead of goals and business value	3 pt	1 pt

## Problem: Determining the true value of the product that the customer needs

Guidelines around which you can improve in your organisation:

Guideline	Feedback from the team
Investigate the users needs in-depth, use both qualitative and quantitative research.	<ul style="list-style-type: none"><li>- Yes, but neither do we have time nor resources -- they are both limited. Time, as we are a tiny team. Resources, as we are a non-core team. Most of the resources are devoted to the core teams.</li></ul>
Collect feedback from users about the product	<ul style="list-style-type: none"><li>- Yes, but I as a Product Designer need the green light to do so (from the Product lead).</li><li>- Yes, we use it, but we don't get a big amount of answers from customers</li><li>- Yes, but we develop part of the product which is used by a small subset of all users. This results in small data samples that makes it difficult to draw valuable insights.</li></ul>
Validate ideas by working on prototypes, close to the customers.	<ul style="list-style-type: none"><li>- Yes, but not too often due to the limited time.</li><li>- Yes, but we have limited access to customers who are willing to participate</li><li>- Yes, but again - a small sample of users make it difficult to experiment on a larger / meaningful scale</li></ul>

**Solutions** that the team perceived as applicable to their projects:

Solution name	I can apply this solution in my project [-6; 6]	This solution is comprehensible [-6; 6]	This solution can solve the problem [-6; 6]
Solution 4: Data analysis and in-depth interviews with clients	6 pt	6 pt	6 pt
Solution 3: In-depth interviews with potential/future users of the product	5 pt	7 pt	4 pt
Solution 1: Customer feedback loop	5 pt	5 pt	4 pt
Solution 2: User testing sessions – testing prototypes [I can apply this solution in my project]	5 pt	5 pt	2 pt
Solution 6: Define Jobs to be done	5 pt	4 pt	3 pt

**Techniques** that the team perceived as could be used in their project:

- Jobs to be done
- Industry Analysis
- Analysis of contact points with customers (i.e. chat, e-mails)
- Value proposition canvas

Key things that the team learned from Software Product Management Guide, that they would like to use in their projects:

- How to define the metrics we want to measure, think about long-term strategy, how to better utilize the sources of user feedback
- New strategies and the importance of gathering user feedback through multiple sources, practices to use in planning with a longer perspective in mind
- the need for centralised user research (make results easily available for all teams),
- longer-term strategy (think 3-5 years ahead) as it's always a good time to start doing this exercise, to gather feedback via surveys / measure satisfaction around JTBD (upon their completion)

## Problem: Balancing between reactive and proactive work

Guidelines around which you can improve in your organisation:

- Guarantee the team a minimum share of proactive work at the level of OKR or team goals.
- Spend time to plan the future – create a vision or north star. Setting product development goals that support this vision and searching for innovations
- Analyse impact and prioritise initiatives or backlog items in order to define these activities that will bring the greatest value to the user (whether they come from a proactive or reactive pool).

**Solutions** that the team perceived as applicable in their projects:

Solution name	I can apply this solution in my project [-6; 6]	This solution is comprehensible [-6; 6]	This solution can solve the problem [-6; 6]
Solution 9: Implementation of maintenance and customer service management standards	4 pt	4 pt	4 pt
Solution 7: Define goals (OKRs) and product roadmap	3 pt	3 pt	3 pt
Solution 6: North star	3 pt	4 pt	4 pt
Solution 5: Customer advisory board	3 pt	4 pt	3 pt
Solution 4: Introduce product discovery process in the organisation	2 pt	4 pt	2 pt
Solution 3: Continuous prioritization (e.g. by RICE method)	2 pt	2 pt	2 pt



**Techniques** that the team perceived as could be used in their project:

- Design Sprint
- Impact Analysis
- Product Goals
- North Star Metric
- OKRs
- Product Vision
- Industry Analysis
- Jobs to be done

Key things that the team learned from Software Product Management Guide, that they would like to use in their projects:

- RICE, OKR, and Moscow analyses - would love to try them out. Value Proposition Canvas - will utilize it. I will also improve our Jobs to be Done.
- New solutions and techniques of balancing work, strategies to implement ways of balancing work on a regular basis