A comprehensive overview of software product management challenges

Olga Springer [1] · Jakub Miler [1]

Abstract

The principal focus of software product management is to ensure the economic success of the product, which means to prolong the product life as much as possible with modest expenditures to maximizs profits. Software product managers play an important role in the software development organization while being responsible for the strategy, business case, product roadmap, high-level requirements, product deployment (release-management), and retirement plan. This article explores the problems that affect the software product management process, their perceived frequency and perceived severity. The data were collected by a systematic literature review (5 main databases were analyzed), interviews (10 software product managers from IT companies), and surveys (89 participants). 95 software product management problems assigned nonexclusively to 7 areas were identified. 27 commonly mentioned software product management problems were evaluated for their perceived frequency and perceived severity. The problems perceived as the most frequent are: determining the true value of the product that the customer needs, strategy and priorities change frequently, technical debt, working in silos, and balancing between reactive and proactive work. In total, 95 problems have been identified which have been narrowed down to 27 problems based on their occurrence in at least 3 interviews. These selected problems were prioritized by perceived frequency and

Communicated by: Tony Gorschek

Highlights
- 7 problem areas in software product management are identified with SLR
- 95 problems related to software product management are identified with interviews
- 27 selected problems are evaluated for their perceived frequency and perceived severity with survey
- 3 problems from earlier research are confirmed, 2 previously mentioned problems are now obsolete
- Many new detailed problems are identified, top related to competencies and technology

✉ Olga Springer
olga.springer@pg.edu.pl

Jakub Miler
jakub.miler@pg.edu.pl

[1] Gdańsk University of Technology, G. Narutowicza 11/12, 80-233 Gdańsk, Poland

🖄 Springer

perceived severity. Some of the identified problems spanned beyond the software product management process itself, but they all affect the work of software product managers.

1 Introduction

The objective of software product management (SPM) is to "prolong product life, as much as possible with modest expenditures to maximize profits, and retain market leadership." Product management responsibilities are crucial in software companies, as these responsibilities support the decision-making process and the development of valuable products (Kittlaus and Fricker 2017). 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 (International Institute of Business Analysis 2015; Project Management Institute (PMI) 2017). It differs from project management in that it focuses more on customers, sales, user feedback and constant product growth. Products are commonly developed without a fixed time limit by a number of projects in a bigger timescale. Agile and Lean approaches share a user-centered perspective and fast business value delivery with the software product management, although they do not cover the more strategic levels of product development. Business analysis covers an important part of product management, but it lacks a technical viewpoint. Software product management effectively spans all of these areas.

Software product management is becoming increasingly integrated into the commonly used software development and management frameworks in the IT industry (CollabNet VersionOne 2019; Schwaber and Sutherland 2017; Scaled Agile Inc. 2019). Scrum supports product management as a framework, as it allows to continuously improve 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 (Schwaber and Sutherland 2017). 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 (Project Management Institute (PMI) 2017; AXELOS 2017). The comprehensive software engineering-related resources do not include software product management as a topic connected to software development (Sommerville 2015). On the other hand, the generic product management and development guidelines and models do not focus on the specific aspects of software as a market product or service (Geracie and Eppinger 2013; CMMI Institute 2019).

The problems and risks related to project management, requirements engineering, Agile, or UX are already widely discussed in the literature (Lopez et al. 2016; Fernández and Wagner 2015; Kagan et al. 2016; Uusitalo et al. 2019). For software product management specifically, the key article focused on the SPM problems is titled "Lean Solutions to Software Product Management Problems" by Maglyas et al. (2012) (Maglyas et al. 2012a). They compiled a short list of 5 quite generic problems, although without any information about their occurrence or scale of impact on product management. This indicates that the research in 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 study (Maglyas et al. 2013; Ebert and Brinkkemper 2014; Maglyas et al. 2017; Jantunen et al. 2013; Bekkers et al. 2012). This is why we found it crucial to understand the problems of software product management more deeply, and to update the information from 2012.

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?

The article is structured as follows. Section 2 provides the theoretical background of our research, the software product management frameworks and processes, the role of a software product manager and an overview of the previously identified software product management problems. In section 3 we describe our research method including a systematic literature review, interviews, and survey. Section 4 reports on the results of the literature review, identifies related work and provides a list of identified software product management problem areas. Next, in section 5 we present the results of the interviews with experienced software product managers who identified 95 software product management problems. Section 6 provides the results of the industrial survey on the 27 problems most commonly mentioned by our experts. We present the evaluation of their perceived frequency and severity by our respondents. In section 7 we discuss our findings, compare them to other papers, and present the implications for theory and practice. Section 8 discusses the risks to the validity of our research and their mitigation. Finally, section 9 presents final conclusions. The interview guide and survey design are attached in Appendices 1 and 2, respectively.

2 Background

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 of the economic success of the product and the company investing in it (Kittlaus and Fricker 2017). It has been shown that the institutionalization of this role measurably improves the success rate of projects (Ebert and Brinkkemper 2014).

2.1 Product management frameworks

The Product Management Lifecycle framework represents the lifespan of a product from conception through retirement (Geracie and Eppinger 2013). The applicable product stages are: development/acquisition, introduction to the market, growth phase, and withdrawal. The framework also defines the activities that happen when a product is already on the market:

commercialization and manufacturing operations, and problems related to software product management may occur throughout the entire product lifecycle. The product management lifecycle is divided into seven phases: conceive, plan, develop, qualify, launch, deliver, and retire. Between phases, there are decision points that are effectively milestones which require critical business decisions. At each decision point, it can be decided to:

1. continue and go on to the next stage,
2. cancel the project,
3. redirect the effort based on strategic changes or gaps,
4. defer a decision.

According to the ProdBoK, a product manager is responsible for the strategy, business case, roadmap, high-level requirements, deployment (release-management), and retirement plan. He or she also collaborates with other teams working on the analysis/design, development, operations, marketing, and sales. It is worth pointing out that the ProdBoK does not take into account the widspread changes the Internet has brought to product marketing and product development. It also does not include and mention any of the Agile practices. (Geracie and Eppinger 2013)

The ISPMA SPM framework defines a set of "core SPM" activities that the software product manager is responsible for most often, which cover product strategy and product planning (Kittlaus and Fricker 2017; ISPMA SPM Framework V.1.3 n.d.). Product strategy includes: positioning and product definition, delivery model and service strategy, sourcing, business case and costing, pricing, ecosystem management, legal and IPR management, and performance and risk management. Product planning includes: product life-cycle management, roadmapping, 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, marketing, sales and distribution, and service and support. They support other teams making sure they perform in line with the product strategy and plan.

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.

Another study reports that a software product manager's role and its responsibilities depend on the company size (Springer and Miler 2018). 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 SPM 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 (Springer and Miler 2018). 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 SPM activities rather than on supporting activities. Companies measure product management based on annual targets such as sales or growth. However, only one third actually have financial KPIs delegated to the product managers (Ebert and Brinkkemper 2014).

Scrum defines a role responsible for business goals – the Product Owner (Schwaber and Sutherland 2017). It may sound similar to the product manager, but the difference is that the spectrum of activities and responsibilities of the product manager is much broader than that of the Product Owner (Kittlaus 2012). Depending on the organization's size, the product manager and the Product Owner roles may be separated or a product manager can act as the Product Owner. In most environments, it makes more sense to have the two roles separated (Kittlaus and Fricker 2017).

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 (Wagenblatt n.d.). 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).

The PYPR framework consist of four steps:

- Step 1: Define, Understand, Structure, and Transparentize (DUST) Product Management Dimensions
- Step 2: Clarify Roles and Responsibilities and Form a (Extended) Product Team
- Step 3: Assess and Visualize the Product Yield Potential
- Step 4: Prioritize and Strategize Actions Based on PYPR Scores Analysis

There is one more study that provides a structured assessment of software product management. The authors of the Maturity Matrix for Software Product Management created a survey which enables product managers to benchmark their organization, assess individual processes and apply best practices to create an effective SPM environment. The main areas covered by the assessment survey are: requirements gathering, requirements identification, requirements organizing, requirements prioritization, release definition, release definition validation, scope change management, launch preparation, core asset roadmapping, product roadmapping, roadmap intelligence, partnering & contracting, and product lifecycle management (Bekkers et al. 2012).

2.2 Problems related to software product management

In this article, we define 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. A problem area is a named set of problems originating from a particular area of the company.

As the frameworks described above show, 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 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 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.

In 2012, Maglyas et al. provided a list of the 5 main problems related to software product management (Maglyas et al. 2012a). 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.
- Problem3: 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, 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.

The contribution from Maglyas et al. remains the only systematic attempt to identify and describe the 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 (Ebert and Brinkkemper 2014). 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.

3 Research method

3.1 Research strategy

The goal of this research is to identify specific problems recognized by Software Product Managers (RQ1) and to evaluate the perceived frequency and perceived severity of the selected problems (RQ2, RQ3) – all from the perspective of software product managers (Table 1).

To achieve the goal of our research, we have followed the pragmatic research philosophy (Goldkuhl 2011), applied mixed methods with the sequential exploratory strategy (Easterbrook et al. 2008), and used a triangulation approach (Walsham 2006) to collect the data on our research topic with several methods and techniques: SLR, interviews and a questionnaire. Our research comprized three steps:

(1)   identification of the problem areas related to software product management,
(2)   identification of the problems related to software product management in particular problem areas,
(3)   evaluation of the perceived frequency and perceived severity of the problems.

3.2 Systematic literature review

The first step aimed at answering RQ1 and involved a review of the current literature using the Systematic Literature Review method (Kitchenham 2004; Dyba et al. 2005). We derived the keywords from RQ1 and selected multiple independent scientific literature databases relevant for computer science together with some aggregate databases. Our inclusion criteria were: conference and journal papers published since 2013 as ProdBoK® (Geracie and Eppinger 2013) was published in that year; publications in English; original research and review papers. We excluded the papers not based on research in software companies (e.g. in the academic environment) and those related to other papers (e.g. extended versions). We also excluded such research subjects as IT hardware, open source software and global software engineering as laying outside of the scope of our research topic. We applied the inclusion criteria directly in the search engines, while the exclusion criteria were applied during the title and abstract review phase. Beside the keyword-based search analysis, we also ran backward snowballing (Wohlin 2014) 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, we screened the papers' titles and abstracts and removed the duplicates. We applied the iterative process 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

Table 1   Research methods and data collection techniques used to answer the research questions

RQ1: What problems are recognized by Software Product Managers?	SLR, interviews
RQ2: What is the perceived frequency of the software product management problems by the software product managers?	questionnaire
RQ3: What is the perceived severity of the software product management problems by the software product managers?	questionnaire

unanimous decision for each paper was reached. For the final data extraction, we selected the papers 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 we are not able to collect from these articles direct problems related to software product management. Because of that, we extracted a list of topics mentioned in the context of software product management problems from each selected paper and categorized them into problem areas using keyword analysis, ground knowledge and our industry experience. The goal was to create an Interview Guide in order to structure the meeting and gain more knowledge in the topics related to software product management.

We defined a topic 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 lets us identify the topic of "continuous integration." We defined a problem area 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, we tagged the articles that included "product management" in the list of keywords, as we wanted to understand how many articles were strictly related to the field that we were researching.

During the analysis, we decided to build our own 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 problem areas and topics were written in British English which was preserved in this article. The SLR was carried out between November 2018 and February 2019, and involved two researchers, who are the co-authors of this article. Further, in 2021 we carried out backward and forward snowballing (Wohlin 2014) and included the results in the field of software product management.

Although not entirely consistent with the SLR guidelines, it is practically acceptable to engage only two researchers (Brereton et al. 2007). The list of problem areas identified through the SLR was used later to construct the interview guide (Appendix 3) and identify the detailed software product management problems from the perspective of software product managers in the next step of the research.

3.3 Interviews

The second step aimed at answering RQ2 with qualitative research. To fill the research gap in the literature, we conducted a series of semi-structured moderated interviews with industry practitioners to identify the problems in software product management from the perspective of the actual software product managers, following the guidelines by Hove and Anda (Hove and Anda 2005). We chose interviews as the research method to facilitate the free identification of problems and keep the scope and depth of research data as open as possible. We employed the semi-structural approach to ensure the coverage of all problem areas from the literature but also to allow the interviewees to mention problems whenever it was convenient to them.

We recruited the participants for the interviews using purposive sampling (Palinkas et al. 2015). We qualified the interviewees by their experience in software product management and the size of the company. We assumed 5 years of experience in software product management to be a sufficient level of expertise regarding software product management problems. We

aimed at interviewing experts from companies of various sizes to cover the specific problems related to the role of a product manager which differs by company size, as shown in our previous research (Springer and Miler 2018). We invited the interviewees using our 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 h. We developed an interview guide in British English (Appendix 2) which 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 by one of the article co-authors.

A list of software product management problems was extracted separately from each interview preserving the original interviewees' statements. We stopped interviews when we identified 10 problems that were mentioned in more than half of the interviews. We carried out inductive coding, 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 responses. After the interview, the work was to convert complex sentences into codes (simplified sentences that are names of problems). When creating the codes, we took into account syntax in order to maintain a similar meaning. To ensure the reliability of the coding, both authors of the research listened to the recordings of the interview, to make sure they understood the problem described by the interviewee. Additionally, the extracted and coded problems were also verified by the interviewees.

Then, the problems were screened for duplicates and identical problems mentioned in different interviews were merged by the article co-authors, in 2 rounds. Combining at this stage consisted in 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. We assigned many problems 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 (Maglyas et al. 2012a; Ebert and Brinkkemper 2014).

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 we 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 we could not further merge without losing their meaningful details.

The problems were written in British English which we preserved in this article. 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 full list of problems with assigned areas can be found in the supplementary data in the online version of this article.

The examples below show how the codes are based on the extracted data.

Example 1
Code: Lack of user analytics

– No access to user behavior data – enterprise customers do not allow it to be tracked,
– Lack of data about user behavior,
– Lack of data analytics,

- No data about product usage,
- Lack of data setup,
- Lack of data analytics on user behavior.

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,
- Stakeholders management – they have different visions and goals,
- If you lack a product strategy then you can not prioritize what you are going to do,
- Lack of a visioner on board.

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, 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.

3.4 Survey

In the third step of our research, to answer research questions RQ2 and RQ3, we chose a quantitative research method – a survey among the software product managers, following the guidelines by Kitchenham and Pfleeger (Kitchenham and Pfleeger 2008). We wanted to evaluate the perceived frequency in which a problem occurred in the software product manager's work as well as the perceived severity of the impact of the problem on the software product manager's work. Compared to other research methods, the survey provided focus on particular problems as well as ease of data collection and analysis with closed questions (Punter et al. 2003). An online survey was built with a dedicated tool of the Gdansk University of Technology (Ankiety 2019). To ensure the survey correctness, we carried out a pilot study with a non-native English speaker and practitioner with 10 years of experience in software

product management. The pilot study resulted in an improved explanation of the survey in the introduction. The final survey was distributed via e-mail, LinkedIn, Facebook, Slack Groups, and internal messaging systems of several IT companies. We used judgment and snowball sampling that resulted in a non-probability (convenience) sample which, however, is typical of online surveys (Fricker 2017). The survey was written in British English.

The total number of identified software product management problems exceeded the capacity of a practical survey, so we had to select a subset of problems for further evaluation to enhance the feasibility and validity of the survey (Punter et al. 2003). We decided to include problems indicated by at least 3 of 10 interviewed experts. To avoid any subjective interpretation of problem importance in this selection, we assumed that the relative importance of the problems can be represented by the number of interviews in which the problem was indicated. These criteria would have resulted in a survey with 60 evaluations at most (30 problems times 2 evaluations each – frequency and severity). Other problems may be investigated in a separate study.

The survey was divided into three parts. In the first part, we asked the demographic questions 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 we presented the respondents 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

and asked them to agree or disagree with these statements in a Likert 5-point response format (strongly disagree, disagree, neutral, agree, strongly agree) (Harpe 2015). 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.

We defined the perceived frequency 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 abovementioned 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 such as ours. We do not refer to the quantitative probability of problem occurrence which could be biased by non-random sampling. Similarly, we defined the perceived severity as the percentage of the product managers that agree that a given problem had a significant impact on their work and calculated it as the total percentage of "agree" and "strongly agree" responses to the abovementioned question (b) on that problem. Again, here we employed an opinion based qualitative definition 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 (Shahin et al. 2019; Murphy-Hill et al. 2015).

Finally, in the third part of the survey, we asked the respondents if they had experienced any other problems in software product management together with the evaluation of their severity, and provided the opportunity to leave an email address to receive the research results.

The survey design is presented in detail in Appendix 3.

4 Results

4.1 Systematic literature review

We used the following search query in the SLR: '("product management" OR "product manager" OR "product owner") AND (software OR IT) AND (problems OR barriers OR limitations OR challenges)'. We searched 5 databases: Elsevier ScienceDirect, Scopus, IEEExplore, SpringerLink, and Web of Science. Table 2 presents the detailed search criteria.

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 backward and forward snowballing, we selected 47 papers for data extraction to identify the topics of the problems, and possibly also some concrete software product management problems. Table 3 presents the number of accepted papers from each database in each SLR step. The number in parentheses shows the number of duplicates removed. For the purpose of snowballing, we scanned around 400 references in the papers from our initial SLR result set, checked about 30 sources and accepted 12 of them. The list of all 47 scanned sources can be found in Appendix 2.

Full-text analysis of the selected papers showed that the majority of the mentioned problems were not directly related to the software product management process and the perspective of the software product manager. This confirms there is a research gap in the field of specific problems related to software product management.

We found out that the reviewed papers named many areas and topics where software product management problems might occur. We proposed 7 problem areas coded A1 to A7 to group similar topics. The list of problem areas and topics shown in Table 4 is the final result of our literature review. The full list of all articles included in the analysis, together with assigned areas and topics for each recognized problem, can be found in Appendix 1.

Table 4 shows that 4 topics that occur the most often in the problems in the literature about software product management are related to: responsibilities (11), stakeholders management (10), strategy (8), and decision-making process (7). Examples of problems from the literature assigned to these topics can be found below.

Responsibilities:
- "Overlapping responsibilities, an overwhelming volume of activities."

Table 2 Detailed search criteria of the SLR

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
IEEExplore	Journals & Magazines	–	2013	Full text
SpringerLink	Articles, Conference paper	Computer science, Engineering, Business management	2013	Title, abstract and keywords
Web of Science	–	–	2013	Full text

Table 3 Number of papers accepted in each SLR step

Database	Total found	Screening title and abstract	Removing duplicates	Field fit scanning
Elsevier ScienceDirect	253	150	150 (0)	20
Scopus	147	81	75 (6)	12
IEEExplore	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

- "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."

Table 4 Problem areas and topics extracted from the analyzed literature (number of appearances in articles tagged as 'SPM related,' number of appearances in the final SLR results)

Id	Problem area	Topics
A1	Requirements engineering	Requirements management (6,13), prioritization (4,7), stakeholders management (10,10), product roadmap (4,5), user involvement (4,7)
A2	Software development process	Technical debt (0,4), continuous integration (1,3), release cycle speed / time-to-market (3,8), software quality (0,5), system architecture (1,4), split testing (0,1), data collection (2,2)
A3	Team	Unqualified people (2,4), motivation (2,2), communication (3,8)
A4	Product knowledge management	Product description (2,2), sharing knowledge (4,6), documentation (2,5)
A5	Organizational – strategic	Strategy (8,7), business model (2,2), market research (1,1), price management (0,1), trust in team (2,2), partnership management (1,3),
A6	Organizational – operational	Tools (2,4), responsibilities (11,12), distributed teams (1,3), continuous experimentation (3,6), decision-making process (7,9), agile (0,3), time (3,7), budget (1,2)
A7	User research	User centered design (3,9), customer feedback (3,9), user behavioral data (1,2), business intelligence (2,3), product design (1,3)

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."

4.2 Interviews

The software product management problems were identified with a set of 10 interviews with practicing software product managers following a predefined interview guide (Appendix 3). Table 5 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.

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. We decided on this threshold to have a feasible number of problems to study in this research. We may decide to investigate other, more rarely mentioned problems in the future. Additionally, we studied the entire list of 95 problems and did not find any new problem areas outside of the selected 27 problems, which shows that the coverage of the software product management scope is satisfactory.

Table 5 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

Table 6 presents the 27 problems identified in at least 3 interviews, together with their assigned problem areas. Additionally, 13 other problems were identified in 2 interviews and 55 problems were identified in 1 interview only.

As table above presents, recognized problems were mentioned by the Interviewees while discussing different areas. For example, "Roadmap focused on features instead of goals and business value" (P51) concerns many areas, as it was mentioned in the 6 of them.

4.3 Survey

We took the 27 software product management problems shown in Table 6 for the evaluation with survey. In total, 333 respondents started the survey. We collected complete surveys from 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 7. Multiple answers were allowed. Their role names were mostly: Product Owner (65.1% of respondents), Product Manager (57.3%), and Head of Product (11.2%). 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.

The distribution of the roles in IT held by the survey respondents anytime in their career is given in Table 8. 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.

The distribution of the size of the respondents' companies is presented in Table 9. 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 our previous research, which showed that dedicated software product managers were being hired when the company started to grow (Springer and Miler 2018).

Figure 1 presents the distributions of the evaluations of perceived frequency for the selected 27 software product management problems. Figure 2 presents the distribution of perceived severity evaluations for the 27 problems;

Table 10 presents the evaluation of the perceived frequency and perceived severity of the 27 software product management problems in our 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 6 Software product management problems identified in at least 3 of 10 interviews

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
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). PM manages requirements instead of identifying problems and opportunities, seeking innovation.	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

Table 6 (continued)

ID	Problem name	Problem areas	No. of interviews
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

Table 7 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%

Table 8 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%

Table 9 Distribution of respondents' company sizes

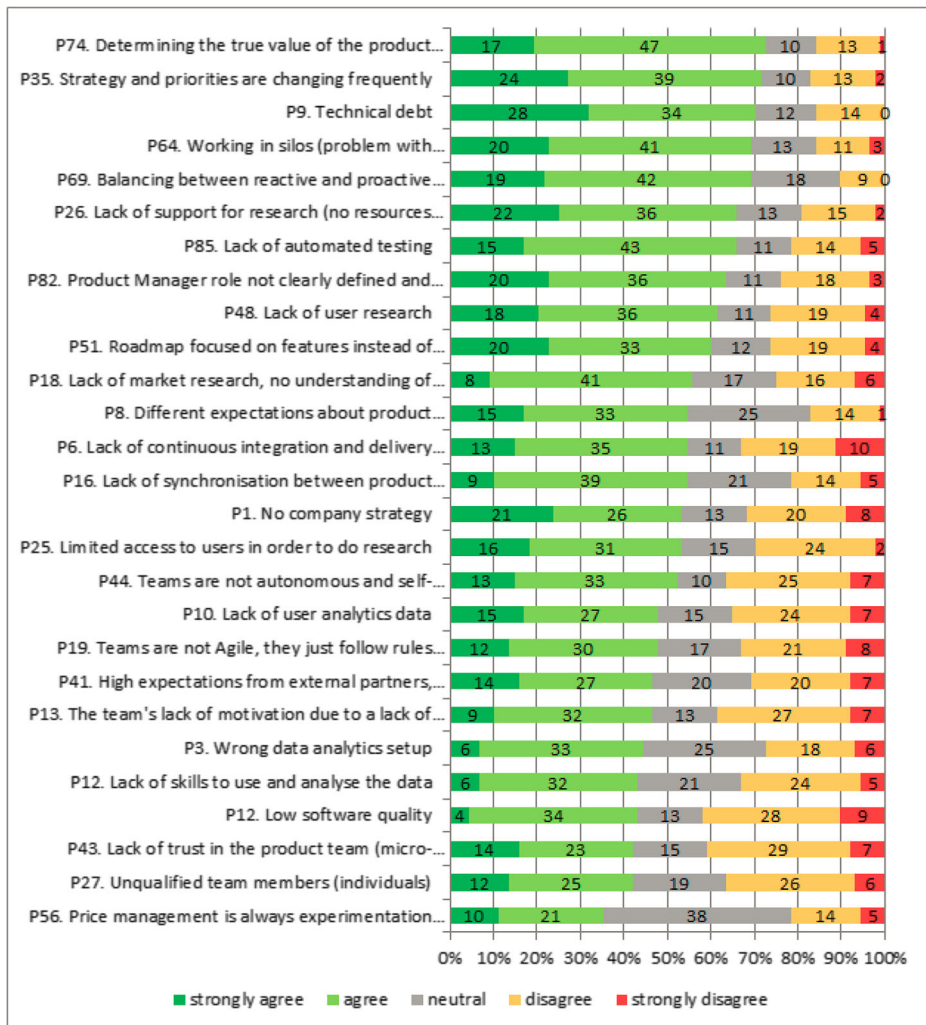| Company size	Number of respondents	% of respondents
1–10	17	19.10%
11–50	29	13.58%
51–250	33	37.07%
>250	49	55.05%

Fig. 1 Distributions of perceived frequency evaluations in the survey responses

The table is ordered first by perceived frequency and second by perceived severity. Figure 3 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 (The Stationery Office 2010; ISO/IEC 31010:2019 2019).

Our research is based on data collected directly from software product managers, and we have not changed the taxonomy of the problem names communicated by the interviewees. Table 10 presents a comprehensive list of the problems from the perspective of the Software Product Managers.

We tested the distributions obtained for the influence of several confounding variables: the role of the respondent, the respondents' country, the respondents company size, and the SPM experience level. We used the Knime tool and the Mann-Whitney U-test.
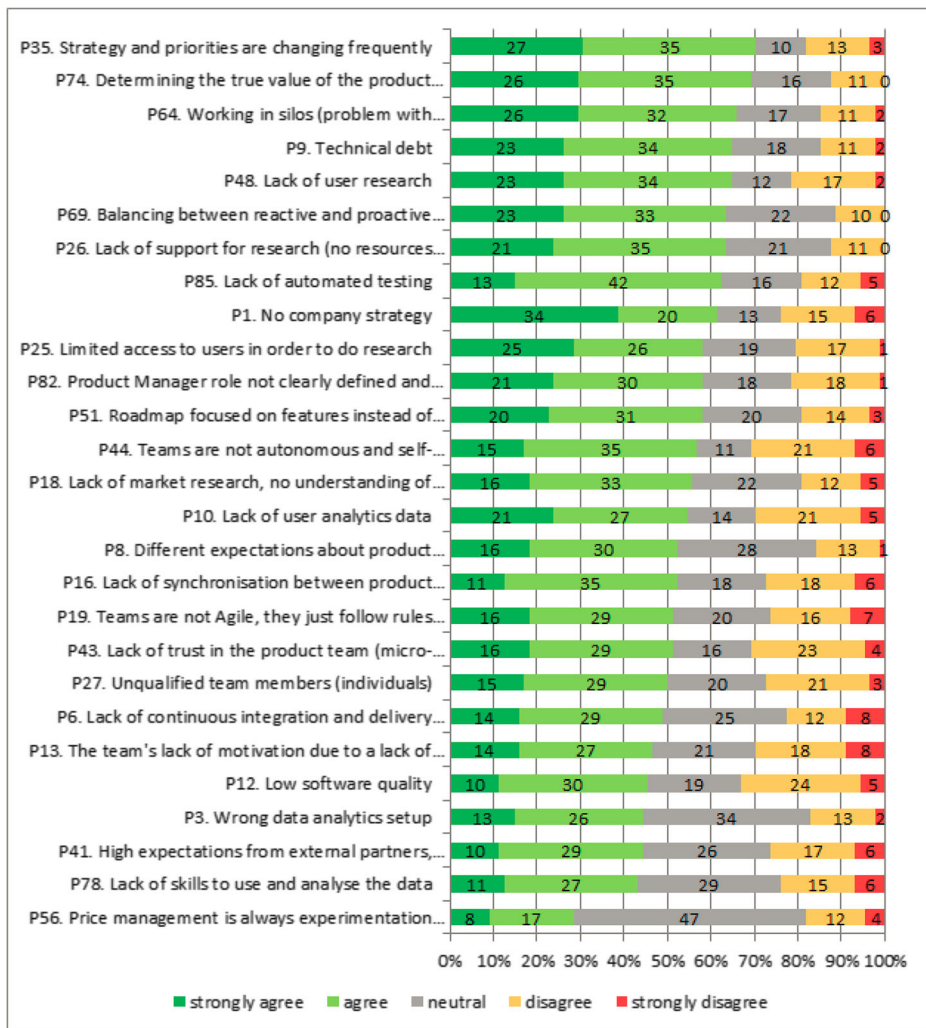
Fig. 2 Distributions of perceived severity evaluations in the survey responses

To test for the role of the respondent, we divided our sample 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, we observed only one statistically significant difference ($p = 0.028$ 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.

To test the respondents' country, we divided our sample into two groups: Poland and Rest of World. We could observe that two problems were perceived as more frequent in Poland that

Table 10 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

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 SPM data analytics competencies in Poland compared to the rest of the world.

To test the company size, we divided our sample into two groups: large companies and micro to medium companies. We 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 =
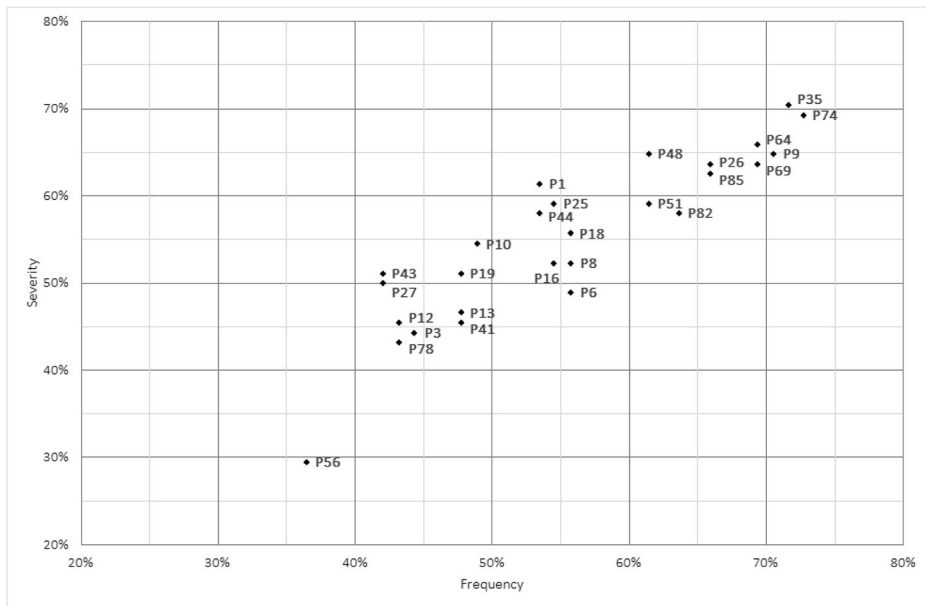
Fig. 3 Evaluation of the software product management problems in the frequency-severity space

0.045, alpha = 0.05). We did not observe any statistically significant differences in the perceived severity between the two groups based on company size. 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 SPM experience, we divided our sample 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.

5 Discussion

5.1 Top frequent problems

The top problem of the software product managers in our survey is to determine the true value of the product that the customer needs (P74). It is a complex problem that requires advanced research to learn what the value is, how to build the product to support this value, and how to ultimately make the product profitable. This problem is the most frequently experienced

(72.7%) and also has a significant impact on the software product manager's work (69.3%). This problem relates to the whole product lifecycle – from introducing to the market until retirement.

A software product manager has a lot of responsibilities and cooperates with many teams. An important problem affecting his or her work is the frequently changing strategy coming from the board (P35). It interrupts previously settled priorities and makes it problematic for software product managers to change the direction and leave what was done before. This problem has the highest impact on the software product manager's work (70.5%) and is also considerably frequent (71.6%).

Another frequent problem is the technical debt (P9) with a perceived frequency of 70.5%. Besides thinking about the value for the customers, their problems and needs, a software product manager has to support the team to resolve the technical problems, which slow down the production and delay releasing new features to the customer. This risk materializes when a product is being developed quickly in new organizations in order to launch it, or there is not enough investment in the technology of the product that is already in use, and the technology is aging.

The next frequent problem (69.3%) is a situation when teams are working in silos divided by competences instead of being multidisciplinary (P64). This is related to the organizational structure of the product and development teams. It raises the problems with communication and requires more time-consuming synchronization between teams, which is problematic for the software product manager.

The fifth most frequent problem is balancing between reactive and proactive work (P69). As the scope of the software product manager's work is very broad, he or she may easily focus more on managing requirements rather than handling research, identifying problems and opportunities.

The most frequent problems described above apply to many areas, which confirms that software product managers have many responsibilities and their problems apply to various aspects of software development.

5.2 The most severe problems

The problems recognized as the most frequent – Determining the true value of the product that the customer needs (P74), Strategy and priorities are changing frequently (P35), Technical debt (P9), and Working in silos (P64) – were also evaluated by the survey participants as the 4 most severe problems, which confirms their overall relative importance.

The fifth most severe problem is Lack of user research (P48). It is very negative for the whole business and product when a product manager does not know their users well. However, this problem is only ninth in terms of perceived frequency, which may indicate that companies more often than not carry out some form of user research.

5.3 The less frequent problems

Of the 27 problems we evaluated with our survey, the problems perceived as the less frequent were related to low software quality (P12), lack of skills to use and analyze data (P78), lack of trust in the product team (P43), unqualified team members (P27) and price management as experimentation burdened with risk (P56). In particular, P56 sticks out and has a significantly lower frequency and severity than the other problems. This may mean that the surveyed

software product managers were not responsible for pricing strategy in the organization, but is may also be caused by the strong qualifying term "always" in the problem name which the respondents could not agree with.

For the less frequent problems, the evaluation of perceived frequency in the survey is consistent with the results of the interviews as these problems were also rarely mentioned in the interviews. P12, P78, P27, P56 were mentioned in 3 interviews only, whereas P27 was mentioned in 4 interviews. No such direct consistency can be observed for the most frequent problems.

5.4 Comparison with problems from the previous research

The problems found in this research can be matched to the activities from the Software Product Management Framework. The problems on our list are related to all 3 groups of software product management activities: "participation," "core," and "orchestration." Similarly to the activities of the framework, there are problems outside of the "core" activities. There are many activities that a Software Product Manager participates in or orchestrates, but he or she is not the owner of them and it is not his or her main responsibility to fix some of the problems. However, in order to build a successful product, an effective software product management process is essential, and there might be a need to influence others to change and fix important problems, even outside of the "core-related" activities.

While analyzing previous research, we haven't recognized comprehensive source of data about software product management problems, nor detailed list of problems from the perspective of software product managers. We found out some problems (Table 11), however, they were distributed between many articles. There are similiarities between problems recognized in this research, and the list below, but the results from our research are more complex - we identified not only more problems (especially related to "orchestration" activities), but also analyzed frequency and severity.

The topics that we recognized as occurring the most often in the literature about product management (responsibilities, decision-making process and strategy) can also be found in the results of our research but named differently (e.g. P35, P82, P44, P1).

The 27 problems on our list show that Software Product Managers struggle with gathering data, requirements, and accessing customers in order to develop valuable solutions (P74, P26, P48, P25, P10). Determining the true value of the product that the customer needs (P74) represents the essence of software product management.

The evaluation of the frequency and severity shows that the problems related to non-core SPM activities that we recognized (i.e. P9, P82, P6) have a significant influence on the work of software product managers, however to check who can solve these problems and how, what the software product manager can do in such a situation, requires separate research. The whole literature review findings prove these problems have not been identified as problems that impact the software product manager.

Table 12 presents the mapping of the 27 problems from our research ordered by perceived frequency to the 5 problems recognized in 2012 by Maglyas et al. Our problems that can be mapped are not exactly the same as the problems from Maglyas et al., however having read the full problem descriptions in article from Maglyas et al., it was found that they belong to the same area and have common themes.

Problem 3 of Maglyas et al., which refers to the collaboration between the organization and the customer, was mapped onto 3 of our problems (P74, P26, and P48). This shows that the

Table 11 List of problems from the literature

L1. The product manager is a person who can influence what happens only by persuasion and motivation.	[S19] (Maglyas et al. 2013)
L2. Responsibility and authority are not shared between people.	[S19] (Maglyas et al. 2013)
L3. Collaboration with other departments including marketing, sales, and development departments.	[S19] (Maglyas et al. 2013)
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.	[S19] (Maglyas et al. 2013)
L5. Limited authority of the product manager.	[S19] (Maglyas et al. 2013)
L6. I get an idea about our product but I do not know how many people are interested in that.	[S19] (Maglyas et al. 2013)
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.	[S19] (Maglyas et al. 2013)
L8. Dysfunctional organization with unclear responsibilities and silo work which results in continuously changing focus and schedules.	[S20] (Ebert and Brinkkemper 2014)
L9. The problem is not so much that the stakeholders are not known, but that the interdependencies, roles and expectations to requirements are unclear.	[S20] (Ebert and Brinkkemper 2014)
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.	[S20] (Ebert and Brinkkemper 2014)
L11. Product managers are mostly subordinate to marketing and play an administrative role, such as maintaining technical roadmaps.	[S20] (Ebert and Brinkkemper 2014)
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.	[S20] (Ebert and Brinkkemper 2014)
L13. A weak perception of the role and its incumbents by senior management, which in turn would not strengthen the role.	[S20] (Ebert and Brinkkemper 2014)
L14. The difficulties with misunderstanding needs, changing and creeping requirements, missed deadlines and budgetary commitments and failing business opportunities.	[S20] (Ebert and Brinkkemper 2014)
L15. No standardized processes across the company with a slow and cumbersome decision-making process and many individual ad hoc agreements.	[S20] (Ebert and Brinkkemper 2014)
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.	[S20] (Ebert and Brinkkemper 2014)
L17. Lack of strategy and unclear strategy and roadmaps with unclear dependencies and fuzzy technical requirements and impacts.	[S20] (Ebert and Brinkkemper 2014)
L18. It is difficult to motivate a team that does not know for whom it is working and that lacks a purpose.	[S25] (Kristinsdottir et al. 2016)
L19. Communication was also mentioned as being one of the Product Owners' main challenges.	[S25] (Kristinsdottir et al. 2016)
L20. Long release cycles, short-term thinking, and a lack of constant collaboration between organizations and customers.	[S34] (Maglyas et al. 2017)
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).	[S41] (Maglyas et al. 2012a)
L22. Collaboration between organizations and customers.	[S41] (Maglyas et al. 2012a)
L23. Short-term thinking.	[S41] (Maglyas et al. 2012a)
L24. Trying to change instantly.	[S41] (Maglyas et al. 2012a)
L25. Strategic and tactical planning are tightly coupled together, but the main problem is the balance between them.	[S42] (Maglyas et al. 2012b)
L26. In SMEs, most decisions were made by top management and product managers acted as advisors and facilitators in solving problems arising between departments.	[S42] (Maglyas et al. 2012b)
L27. Issues related to roadmapping, and especially to aligning problems with business goals.	[S43] (Maglyas et al. 2012c)
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.	[S43] (Maglyas et al. 2012c)

Table 12 Software product management problems found in our research mapped onto problems of Maglyas et al. using recognized common themes

Rnk	Our research	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
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. When comparing hypotheses with facts, hypothesis loses in value to facts (such as clients' requests, bugs). Managing requirements instead of identifying problems and opportunities, seeking innovation.	–	–
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)	–	–
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	–	–

problem as a whole still exists and has an impact on software product management. The high perceived frequency and severity of our 5 problems also confirms this.

Problem 4 (Short-term thinking) of Maglyas et al. was mapped to 3 of our problems, which also suggests there is room for improvement when it comes to strategy-related work.

Problem 1 (Long Release Cycle) of Maglyas et al. was mapped to 2 of our problems, which also suggests there is a room for improvement for structuring the teams using different methods, to avoid silos and units, which slows down the development process.

Problem 2 (No metrics for evaluating work) of Maglyas et al. was mapped to 1 of our problems, which says the problem still occurs, however when we look at the perceived frequency analyzed in this research, this problem is not common, which may mean that since 2012, companies have learned that measuring the performance of the software product management process is an important factor of company success.

The only problem of Maglyas et al. that has no coverage in our survey is problem 5: Trying to change instantly. This may mean that since 2012, organizations have learned how to introduce software product management processes in the company and that this is not so painful anymore.

We also recognized 18 new problems in our research that we could not be matched with any of the problems of Maglyas et al.

Technology-related problems (P9, P85, P6, P12) were not mentioned in 2012, as probably at that time companies were not suffering from technical debt just yet and the techniques such as automated testing and continuous integration were not yet recognized by software product managers to improve product development and reduce time-to-market. The software product management problems related to competencies might have emerged recently due to the very rapid development of the IT market and the failure of academia and training to keep up with these needs, which results in a continuous lack of competent IT staff. We suspect that problems related to attitudes might not have appeared in the research of Maglyas et al. because of cultural differences. Although both studies focused on the European market, the research of Maglyas et al. was carried out in Scandinavia, while ours mostly in Poland. The lack of motivation and trust might be culture-specific. The problems in stakeholder management may have increased recently due to the increasing scope of software product management, number and diversity of stakeholders involved and high expectations towards product managers. The problems of user and data analytics are related to the emergence of big data, increasing dependence on data and data-driven management. These areas were not as developed and prominent in 2012 as they are today.

The problems SPMs face have evolved since they were researched for the first time in 2012, and it can be observed that 2 out of 5 problems from 2012 have decreased in importance while new problems have emerged. The most frequent and severe new problems are: technical debt, lack of automated testing, different expectations about product management communication per stakeholder, and lack of continuous integration and delivery. The other 10 new problems were ranked at positions 18 to 27, although they still had about 45% perceived frequency.

5.5 Implications for theory and practice

A list of problem areas and topics in software product management is proposed. The list can be used to investigate more aspects of software product management in further research.

The main outcome of our work is the list of problems that can also be used for further research. For example, it may be studied if there is any correlation between problems,

company size, stage of the product lifecycle and other variables. Research can also be conducted with the top management to check what their perspective is and how they perceive problems related to software product management, as it affects their business. Another interesting question is which of these problems are the responsibility of software product managers, and which are not. The perspective of software product managers is important, and hopefully new frameworks to solve the problems will emerge in the future.

This study forms a theoretical basis for further research of software product management problems. The problems can be analyzed for mutual relations, coexistence, hierarchy, causality, etc. The solutions to these problems might be searched for and could be further evaluated for their effectiveness and effort. Additionally, the problems and solutions might be studied in different contexts of applicability such as company size, product type (B2B, B2C), and product lifecycle phase. Finally, a ranking of solutions might be elaborated.

Results from the research can potentially be used by the authors of existing SPM assessment frameworks, while performing evaluations and making adjustments. The recognized problems may also be used to create new assessment frameworks: the SPM process itself or the maturity of the organization, as problem mitigation can help increase the maturity of the organization.

Our work also contributes to the field by developing a survey tool for researchers, which investigates the perceived frequency and perceived severity of the problems. Using this research instrument, more problems can be verified and the list of common and severe problems may be extended. Our findings extend the existing knowledge on the frequency and severity of the problems related to software product management.

The results of the research provide a valuable practical overview of the possible weaknesses of the software product management process in an IT company along with the pains of software product managers. This research provides a broader view of software product management and the factors that need to be considered while setting up the product department or team and planning its development.

The presented list of problems together with their relative rankings may be particularly useful by inexperienced software product managers as they may not yet have faced many of the problems in their career. Also, the problems may differ from one company to the other, so software product managers with experience only in one company may also benefit from reviewing the list of common problems as a kind of checklist to verify their own situation.

The knowledge may be used not only by software product managers, but also by team managers and company owners. Our study describes problems from the perspective of practising product managers that have an impact on the entire company business, which is the responsibility of company owners. We aim at studying the solutions to these problems in our future research.

6 Threats to validity

We conducted a methodical analysis of threats to the validity of our research separately for the three stages of the research.

6.1 Systematic literature review

The analysis of threats to the validity of the SLR follows a map of the threats by Zhou et al. (Zhou et al. 2016).

The threat of non-specification of the SLR's configuration and sufficient details was mitigated with a 10-step Systematic Literature Review plan which included mutual verification and review of the partial results by both researchers, as described in Section 3.2. In each step, one of the authors played the role of the primary researcher and the other the role of the reviewer. We switched roles between steps and split the processing of partial results to increase the validity of the final judgments. We have explicitly documented each SLR step.

The threat of incorrect or incomplete search terms in the automatic searches was mitigated as we identified a set of natural language keywords derived from our research questions. We discussed them and agreed on a set of synonyms to be used in the search query, which mitigated the threat of a lack of standard languages and terminologies. The term "product management" combined with "software" and "problems" are assumed to represent our area of research.

We mitigated the threats of an inappropriate search method and restricted time span using forward and backward snowballing which let us find relevant papers from before 2013 and after 2019. To mitigate the threat of incomprehensive databases, we searched 5 scientific databases related to computer science, 2 of which (Scopus and Web of Science) aggregate papers from many sources.

The threats of inappropriate inclusion & exclusion criteria, as well as inappropriate research questions, were mitigated by the fact that the criteria were defined by two researchers, both experienced in the field of software product management. The research question and the criteria were explicitly documented in an SLR plan.

The threat of inadequate size and number of samples was mitigated by tuning the search query and criteria to reach a sample size of about 1000 records, which is a considerable size for only two researchers to analyze. The culture bias threat was mitigated by using the English language in our search.

The threat of bias in study selection was mitigated by filtering the papers by inclusion and exclusion criteria which involved 3 steps of independent evaluation by each author, discussion of the differences, and reevaluation. In each step, we agreed on the evaluation of about 70% of the papers in question. The other 30% was further discussed and reevaluated in subsequent steps until we reached full agreement on the evaluation for all papers.

The threat of an inaccessible paper/database was mitigated by using the paid, licensed access to the selected databases provided by the Gdansk University of Technology. The threat of primary study duplication was mitigated as we conducted the phase of duplicate removal using DOI as the unique identifier of each paper. The threats of bias in data extraction and subjective interpretation of the extracted data were mitigated, as the extraction of the problems and the problem areas from the final papers was carried out independently by both authors and merged into the final list following a discussion.

Following the well-defined Systematic Literature Review process (Dyba et al. 2005) we confirmed in the literature review that there is a research gap related to the identification and evaluation of the problems of software product managers.

6.2 Interviews

As there is no formal definition of product management, the scope of the interviews may have not covered some of its aspects. However, we included all problem areas in the interview guide that are found in the literature. The interviews asked open-ended questions. The interviews were conducted in natural language (8 of them in Polish, 2 in English). To further increase the

understanding of the topic by the interviewees, the interview guide was sent to them by e-mail prior to the interviews.

The interviews were performed with individually invited practitioners known earlier to the article authors. They were informed about the research goal and sent the interview guide, but no problems were identified prior to the interview. Additionally, they were assured that the identified problems would be attributed neither to their company nor to them personally, which should increase their honesty and trust. The data were collected through open-ended questions to mitigate the interviewer (confirmation) bias. Further, to mitigate the effect of fatigue of the interviewees, the ordering of the problem areas asked was different in each interview. The interviews were recorded and the data was extracted and coded based on these recordings, listened to several times, if necessary. The coded problems were validated with the interviewees. The results of each interview were coded separately and only then merged together. The merging and grouping of the coded problems engaged both paper authors.

The identified problems may be biased towards large enterprises as 7 out of 10 interviewees worked in such companies, but prior to that they worked in smaller companies and the interview was not limited to their current place of work. Additionally, methodical product management is mostly observed in bigger companies, as shown by previous studies (Ebert and Brinkkemper 2014; Springer and Miler 2018).

6.3 Survey

To analyze the threats to the validity of the survey, we followed the list of survey design problems from Ghazi et al. (Ghazi et al. 2019).

To mitigate the threat of Insufficient Sample Size, we used a non-systematic sampling method (group and personal invitations through various social media), thus the sample may not evenly represent the entire population. However, our respondents came from companies of different sizes, held different roles throughout their careers and occupied different product management posts. On average, they had more than 3 years of software product management experience. Our sample size was limited to 89 due to several factors. Many respondents could not complete the survey due to its considerable size and the time required. The most effective distribution channel, LinkedIn, was heavily based on our prior business relations, which were also limited by the time of this research. Additionally, we could not offer any incentives to increase participation in the survey.

The Likert Scale Problems were mitigated with the 5-point Likert response format (Harpe 2015), which allowed for a neutral answer. This scale is subjective, thus we can only draw conclusions on the perceived frequency and severity of the problems, not the real ones.

The threat of Flaws in the Wording of Questions was mitigated with the design process. The survey included 27 problems expressed in natural language. The names of these problems were taken from the interviews. The coded problems were validated with the interviewees, who generally confirmed their correctness and rarely proposed minor changes. Some problems presented in the survey combined several problems from the interviews. In such cases, the final expressions were independently verified by both paper authors for consistency with the original problems. The initial survey questions about respondents' role, experience, company, etc., allowed the respondents to judge their level of understanding of the research topic even further. The respondents answered the survey in a well-known Likert response format (Harpe 2015). To ensure that the survey was well written, correct, and understandable by non-native

English speakers, we conducted a pilot study with a non-native English speaker, which resulted in an improved introductory explanation of the survey.

We mitigated the threat of Invalid Responses by making the survey voluntary. We had limited control over collecting the responses in our survey. We do not know the number of people that received or read the invitations. The invitations were sent to product management interest groups on social media as well as personal contacts of the authors. We also described the topic and goal of the survey on the welcome page and asked several qualifying questions about the software product manager role and experience. We experienced the threat of High Drop-Out Rates. In total, 333 respondents started the survey, but only 89 completed it, which may indicate that a high level of competence and commitment was necessary. Although we cannot exclude that a non-eligible person filled in the interview, we consider it highly unlikely.

There was a technical possibility to fill in the survey more than once, however we could not detect any signs of such an event in the data. Due to the effort needed to complete the survey, again, we consider multiple entries from the same person highly unlikely.

The trustworthiness and commitment of the respondents poses a threat to any survey research, and online surveys in particular. We could not find any reasonable rationale to justify potential malevolent influence on the survey results from the respondents. The survey addressed a topic potentially interesting to the respondents and was shared among the software product management interest groups. 22 of the respondents left their email addresses, further expressing their interest in the results of this study.

We do not know the historical events that could have determined the respondents' perception of the frequency and severity of the software product management problems. As the survey questions were expressed in present-tense, the respondents should have focused on their current company or project. The questions were not put in any specific context of a company, project, business domain, etc.

The threat of Biases Due to Question-Order Effect while filling out the survey could not be mitigated as the ordering of the questions was not randomized. We had to ask about the evaluation of the software product management problems in a fixed order due to limitations of the dedicated survey tool of the Gdansk University of Technology (Ankiety 2019).

The threat of the Common Biases of Respondents could not be entirely eliminated. It must be noted that the survey results represent the subjective view of the respondents and reflect the perceived frequency and severity of the software product management problems instead of the objectively measurable ones. However, our general findings are consistent with prior research (Kittlaus and Fricker 2017; Maglyas et al. 2012a; Ebert and Brinkkemper 2014), but include a more detailed list of problems.

The survey provided 89 evaluations of 27 selected problems. We cannot draw any conclusions on the frequency and severity of the other identified problems.

7 Conclusions

The goal of the research was to identify and evaluate the problems of software product management from the perspective of software product managers. The main contribution of this article is the list of 95 problems related to software product management assigned to 7 problem areas together with the evaluation of the perceived frequency and perceived severity of the selected 27 problems, which answers all research questions (RQ1, RQ2, RQ3). With the Systematic Literature Review, we identified 7 areas of problems related to software product

management. Next, with a series of 10 interviews with professional software product managers, we identified 95 unique software product management problems in these 7 areas (RQ1). Finally, we conducted an online survey from which we obtained an evaluation of the perceived frequency and perceived severity of the selected 27 software product management problems mentioned in at least 3 interviews (RQ2, RQ3).

The software product management problems identified in our research are more detailed and specific than those described in previous research (Maglyas et al. 2012a). Additionally, we identified at least 14 new problems, 4 of which have relatively high perceived frequency and severity: technical debt, lack of automated testing, different expectations about product management communication per stakeholder, and lack of continuous integration and delivery. We have provided an evaluation of the perceived frequency and severity of the 27 software product management problems, both known and new. The whole 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. The resulting list of 27 unique software product management problems should be further researched, especially in terms of how companies, top managers, product managers and other stakeholders can reduce their frequency and impact on software product management.

The resulting list of 27 unique software product management problems will be further studied within our research project, especially in terms of how product managers can reduce the frequency of these problems and their impact on software product management.

Appendix 1: Topics and problem areas from SLR

This data can be found online: *(currently sent as a supplementary material)*

Appendix 2: List of articles from SLR

S1   Juliana Medeiros, Alexandre Vasconcelos, Carla Silva, Miguel Goulão, Quality of software requirements specification in agile projects: A cross-case analysis of six companies, Journal of Systems and Software, Volume 142, 2018, Pages 171–194, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2018.04.064.

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, Journal of Systems and Software, Volume 124, 2017, Pages 82–103, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2016.11.018.

S3   George Valença, Carina Alves, Slinger Jansen, Strategies for managing power relationships in software ecosystems, Journal of Systems and Software, Volume 144, 2018, Pages 478–500, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2018.07.036.

S4   Stefan Wiesner, Sara Nilsson, Klaus-Dieter Thoben, Integrating Requirements Engineering for Different Domains in System Development – Lessons Learnt from Industrial SME Cases, Procedia CIRP, Volume 64, 2017, Pages 351–356, ISSN 2212–8271, https://doi.org/10.1016/j.procir.2017.03.013.

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, Information and Software Technology, Volume 105, 2019, Pages 78–94, ISSN 0950–5849, https://doi.org/10.1016/j.infsof.2018.08.006.

S6   Nirnaya Tripathi, Eriks Klotins, Rafael Prikladnicki, 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, Journal of Systems and Software,

Volume 146, 2018, Pages 130–151, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2018.08.059. (https://www.sciencedirect.com/science/article/pii/S0164121218301729)

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, Journal of Systems and Software, Volume 133, 2017, Pages 195–211, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2017.07.009.

S8   Suprika Vasudeva Shrivastava, Urvashi Rathod, A risk management framework for distributed agile projects, Information and Software Technology, Volume 85, 2017, Pages 1–15, ISSN 0950–5849, https://doi.org/10.1016/j.infsof.2016.12.005.

S9   Daniel Ståhl, Torvald Mårtensson, Jan Bosch, The continuity of continuous integration: Correlations and consequences, Journal of Systems and Software, Volume 127, 2017, Pages 150–167, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2017.02.003.

S10   Brian Fitzgerald, Klaas-Jan Stol, Continuous software engineering: A roadmap and agenda, Journal of Systems and Software, Volume 123, 2017, Pages 176–189, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2015.06.063.

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, Procedia CIRP, Volume 72, 2018, Pages 973–978, ISSN 2212–8271, https://doi.org/10.1016/j.procir.2018.03.151.

S12   Eero Laukkanen, Juha Itkonen, Casper Lassenius, Problems, causes and solutions when adopting continuous delivery—A systematic literature review, Information and Software Technology, Volume 82, 2017, Pages 55–79, ISSN 0950–5849, https://doi.org/10.1016/j.infsof.2016.10.001.

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, Information and Software Technology, Volume 96, 2018, Pages 141–160, ISSN 0950–5849, https://doi.org/10.1016/j.infsof.2017.11.015.

S14   Jesse Yli-Huumo, Andrey Maglyas, Kari Smolander, How do software development teams manage technical debt? – An empirical study, Journal of Systems and Software, Volume 120, 2016, Pages 195–218, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2016.05.018.

S15   Eveliina Lindgren, Jürgen Münch, Raising the odds of success: the current state of experimentation in product development, Information and Software Technology, Volume 77, 2016, Pages 80–91, ISSN 0950–5849, https://doi.org/10.1016/j.infsof.2016.04.008.

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, Journal of Systems and Software, Volume 103, 2015, Pages 267–280, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2014.09.038.

S17   Sami Jantunen, Donald C. Gause, Using a grounded theory approach for exploring software product management challenges, Journal of Systems and Software, Volume 95, 2014, Pages 32–51, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2014.03.050.

S18   Meghann L. Drury-Grogan, Performance on agile teams: Relating iteration objectives and critical decisions to project management success factors, Information and Software Technology, Volume 56, Issue 5, 2014, Pages 506–515, ISSN 0950–5849, https://doi.org/10.1016/j.infsof.2013.11.003.

S19   Andrey Maglyas, Uolevi Nikula, Kari Smolander, What are the roles of software product managers? An empirical investigation, Journal of Systems and Software, Volume 86, Issue 12, 2013, Pages 3071–3090, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2013.07.045.

S20   Christof Ebert, Sjaak Brinkkemper, Software product management – An industry evaluation, Journal of Systems and Software, Volume 95, 2014, Pages 10–18, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2013.12.042.

S21   Koczkodaj, W.W., Dymora, P., Mazurek, M., Strzałka, D., Consistency-driven pairwise comparisons approach to software product management and quality measurement, Advances in Intelligent Systems and Computing, Volume 761, 2019, Pages 292–305, 13th International Conference on Dependability and Complex Systems, DepCoS-RELCOMEX 2018; Lwowek Slaski; Poland; 2 July 2018 through 6 July 2018; Code 213819

S22   Lin, C. Data driven product management, IEEE Engineering Management Review, Volume 46, Issue 1, 1 March 2018, Pages 16–18

S23   Heikkilä, V.T., Paasivaara, M., Lasssenius, C., Damian, D., Engblom, C. Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson, Empirical Software Engineering, Volume 22, Issue 6, 1 December 2017, Pages 2892–2936

S24   Paasivaara, M., Lassenius, C. Scaling scrum in a large globally distributed organization: A case study, Proceedings – 11th IEEE International Conference on Global Software Engineering, ICGSE 2016, 26 September 2016, Article number 7577422, Pages 74–83, 11th IEEE International Conference on Global

Software Engineering, ICGSE 2016; Irvine; United States; 2 August 2016 through 5 August 2016; Category numberE5895; Code 124102

S25   Kristinsdottir, S., Larusdottir, M., Cajander, Å., Responsibilities and challenges of product owners at Spotify – an exploratory case study, Volume 9856 LNCS, 2016, Pages 3–16, IFIP WG 13.2/13.5 Joint 6th International Conference on Human-Centered Software Engineering, HCSE 2016 and 8th International Conference on Human Error, Safety, and System Development, HESSD 2016; Stockholm; Sweden; 29 August 2016 through 31 August 2016; Code 180489

S26   Morris, A., Setchi, R., Prickett, P. Product change management and future information architectures, Smart Innovation, Systems and Technologies, Volume 52, 2016, Pages 241–250, 3rd International Conference on Sustainable Design and Manufacturing, SDM 2016; Chania; Greece; 4 April 2016 through 6 April 2016; Code 173399

S27   Oliveira, F., Goldman, A., Santos, V., Managing Technical Debt in Software Projects Using Scrum: An Action Research, Proceedings – 2015 Agile Conference, Agile 2015, 29 September 2015, Article number 7284597, Pages 50–59, Agile Conference, Agile 2015; National Harbor; United States; 3 August 2015 through 8 August 2015; Category number E5555; Code 118294

S28   Lucassen, G., Dalpiaz, F., Werf, J.M.V.D., Brinkkemper, S., Bridging the Twin Peaks -- The Case of the Software Industry, Proceedings – 5th International Workshop on the Twin Peaks of Requirements and Architecture, TwinPeaks 2015, 10 August 2015, Article number 7184709, Pages 24–28, 5th International Workshop on the Twin Peaks of Requirements and Architecture, TwinPeaks 2015; Florence; Italy; 17 May 2015 through; Category numberE5470; Code 116887

S29   Bosch, J. Architecting to Ensure Requirement Relevance: Keynote TwinPeaks Workshop, Proceedings – 5th International Workshop on the Twin Peaks of Requirements and Architecture, TwinPeaks 2015 10 August 2015, Article number 7184704, Pages 1–2, 5th International Workshop on the Twin Peaks of Requirements and Architecture, TwinPeaks 2015; Florence; Italy; 17 May 2015 through; Category numberE5470; Code 116887

S30   Lehtinen, T.O.A., Virtanen, R., Heikkilä, V.T., Itkonen, J., Why the development outcome not meet the product owners' expectations? Lecture Notes in Business Information Processing Volume 212, 2015, Pages 93–104, 16th International Conference on Agile Software Development, XP 2015; Helsinki; Finland; 25 May 2015 through 29 May 2015; Code 142639

S31   Fabijan, A., Holmström, H., Bosch, J. Customer feedback and data collection techniques in software R&D: A literature review, Lecture Notes in Business Information Processing, Volume 210, 2015, Pages 139–153, 6th International Conference on Software Business, ICSOB 2015; Braga; Portugal; 10 June 2015 through 12 June 2015; Code 119399

S32   Jantunen, S., Hietaranta, K., Gause, D.C. Adjusting to increasing product management problems: Challenges and improvement proposals in one software company, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Volume 7830 LNCS, 2013, Pages 386–400, 19th International Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ 2013; Essen; Germany; 8 April 2013 through 11 April 2013; Code 96369

S33   Komssi, M., Kauppinen, M., Töhönen, H. et al. Roadmapping problems in practice: value creation from the perspective of the customers. Requirements Eng 20, Pages 45–69 (2015). https://doi.org/ 10.1007/s00766-013-0186-3

S34   Maglyas, Andrey & Nikula, Uolevi & Smolander, Kari & Fricker, Samuel. (2017). Core Software Product Management Activities. Journal of Advances in Management Research. 14. 23–45. DOI 10.1108/JAMR-03-2016-0022.

S35   Fotrousi, Farnaz & Izadyan, Katayoun & Fricker, Samuel. (2013). Analytics for Product Planning: In-Depth Interview Study with SaaS Product Managers. DOI 10.1109/CLOUD.2013.33.

S36   Sebastian Barney, Aybüke Aurum, Claes Wohlin, A product management challenge: Creating software product value through requirements selection, Journal of Systems Architecture, Volume 54, Issue 6, 2008, Pages 576–593, ISSN 1383–7621, https://doi.org/10.1016/j.sysarc.2007.12.004.

S37   Christof Ebert, The impacts of software product management, Journal of Systems and Software, Volume 80, Issue 6, 2007, Pages 850–861, ISSN 0164–1212, https://doi.org/10.1016/j.jss.2006.09.017.

S38   Steinhardt, Gabriel. (2017). The Product Manager's Toolkit®. DOI 10.1007/978-3-319-49,998-7.

S39   Ebert, Christof. (2014). Software Product Management. Software, IEEE. 31. 21–24. DOI 10.1109/MS.2014.72.

S40   Hans-Bernd Kittlaus, Samuel A. Fricker (2017), Software Product Management. The ISPMA-Compliant Study Guide and Handbook

S41   Maglyas, Andrey & Nikula, Uolevi & Smolander, Kari. (2012). Lean Solutions to Software Product Management Problems. Software, IEEE. 29. 40–46. DOI 10.1109/MS.2012.108.

S42 Maglyas, Andrey & Nikula, Uolevi & Smolander, Kari. (2012). Comparison of Software Product Management Practices in SMEs and Large Enterprises. Lecture Notes in Business Information Processing. 114. 15–26. DOI 10.1007/978-3-642-30,746-1_2.

S43 A. Maglyas, U. Nikula, and K. Smolander, "What do practitioners mean when they talk about product management?" 20th IEEE International Requirements Engineering Conference (RE), 2012, pp. 261–266. DOI 10.1109/RE.2012.6345812

S44 Maglyas, Andrey & Fricker, Samuel. (2014). The Preliminary Results from the Software Product Management State-of-Practice Survey. DOI: 10.1007/978-3-319-08738-2_21.

S45 Weerd, I. van de; Brinkkemper, S.; Nieuwenhuis, R.; Versendaal, J.M.; Bijlsma, A. (2006), A reference framework for software product management, Technical report UU-CS, volume 2006, issue 014

S46 E. Kim "User-Centered Design Roadmapping: Anchoring Roadmapping in Customer Value Before Technology Selection," IEEE Transactions on Engineering Management, DOI: 10.1109/TEM.2020.3030172.

S47 Wagenblatt, Timo. (2019). Software Product Management: Finding the Right Balance for YourProduct Inc. DOI 10.1007/978-3-030-19,871-8.

S48 Saltan, Andrey & Jansen, Slinger & Smolander, Kari. (2018). Decision-making in Software Product Management: Identifying Research Directions from Practice.

Appendix 3: Interview Guide

General questions and company context

1. General information about the interviewee (name, company, e-mail address, phone number, position, experience).
2. General information about the company (name, number of people, number of IT people, work area, goals).
3. Could you describe the management hierarchy of the company?

Software product management team

1. Could you describe the responsibilities of the product team?
2. How many people are part of the product team?
3. What roles do you have in the product team?

Main problems related to software product management

1. What areas of software product management have you found as the most problematic?
2. What problems have you experienced/observed related to **Requirements Engineering**? (Prompts: requirements management. Prioritisations, stakeholders management, product roadmap, user involvement, etc.)
3. What problems have you experienced/observed related to the **Software development process**? (Prompts: technical debt, continuous integration, release cycle speed / time-to-market, software quality, system architecture, split testing, data collection, etc.)
4. What problems have you experienced/observed related to the **Team**? (Prompts: unqualified people, motivation, communication, etc.)
5. What problems have you experienced/observed related to **Product knowledge management**? (Prompts: product description, documentation, etc.)

6. What problems have you experienced/observed related to **Organizational – strategic**? (Prompts: strategy, business model, market research, price management, trust in team, partnership management, etc.)
7. What problems have you experienced/observed related to **Organizational – operational**? (Prompts: tools, responsibilities, distributed teams, continuous experimentation, decision-making process, agile, time, budget, etc.)
8. What problems have you experienced/observed related to **User research**? (Prompts: user centred design, customer feedback, continuous feedback, usability work, user behavioral data, business intelligence, product design, etc.)

Are there any other problems that you have experienced that cannot be classified to any of the groups above?

Appendix 4: Survey design

Part I:

1. Country: (input)
2. Professional experience in IT in years: (input)
3. Roles in IT you have held at any time in your career: (checkboxes, multiple-choice)

• Administrator
• Analyst
• Architect
• Designer
• DevOps
• Programmer
• Product manager
• Project manager
• Tester
• Validation Engineer
• Scrum master
• Other ……

4. Professional experience as a Software Product Manager in years: (input)
5. What is/was the name of the role where you are/were responsible for software product management? (checkboxes, multiple choice)

• Software Product Manager
• Product Manager
• Product Owner
• Head of Product
• Chief Product Officer
• Other (input)

6. Size of the companies that you worked for as a Software Product Manager (checkboxes, multiple choice):

- 1–10
- 10–50
- 50–250
- 250+

Part II:
Problem name: <name of the subsequent problem from 1 to 27>.

I experience this problem in my work as a product manager.
Likert 5-point scale: strongly disagree, disagree, neutral, agree, strongly agree.

This problem has a significant impact on my work as a product manager.
Likert 5-point scale: strongly disagree, disagree, neutral, agree, strongly agree.

If you have experienced any other problems you want to share with us, please list them below adding your evaluation of their impact on your work (input).

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10664-022-10134-5.

Declaration

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Ankiety PG (2019) GUT CUI https://ankiety.pg.edu.pl/?lang=en. Accessed 23 April 2020
AXELOS (2017) Managing Successful Projects with PRINCE2® (6th edn)
Bekkers W, Brinkkemper S, van Den Bemd L, Mijnhardt F, Wagner C, van De Weerd I (2012) Evaluating the software product management maturity matrix. In: 2012 20th IEEE Int Requir Eng Conf RE 2012 – Proc, pp. 51–60. https://doi.org/10.1109/RE.2012.6345839

Brereton P, Kitchenham BA, Budgen D, Turner M, Khalil M (2007) Lessons from applying the systematic literature review process within the software engineering domain. J Syst Softw 80:571–583. https://doi.org/10.1016/j.jss.2006.07.009

CMMI Institute (2019) CMMI(R) V2.0 Model at-a-glance

CollabNet VersionOne (2019) The 13th annual STATE OF AGILE report – 2018, CollabNet VersionOne

Dyba T, Kitchenham BA, Jorgensen M (2005) Evidence-based software engineering for practitioners. IEEE Softw 22:58–65. https://doi.org/10.1109/MS.2005.6

Easterbrook S, Singer J, Storey M-A, Damian D (2008) Selecting empirical methods for software engineering research. In: Shull F, Singer J, Sjøberg DIK (eds) Guid To Adv Empir Softw Eng. Springer. https://doi.org/10.1007/978-1-84800-044-5_11

Ebert C, Brinkkemper S (2014) Software product management – an industry evaluation. J Syst Softw 95:10–18. https://doi.org/10.1016/j.jss.2013.12.042

Fernández DM, Wagner S (2015) Naming the pain in requirements engineering: A design for a global family of surveys and first results from Germany. Inf Softw Technol Elsevier BV 57:616–643. https://doi.org/10.1016/j.infsof.2014.05.008

Fricker RD (2017) Sampling methods for online surveys. In: Fielding NG, Lee RM, Blank G (eds) Handb Online Res Methods. SAGE, pp 162–183. https://doi.org/10.4135/9781473957992.n10

Geracie G, Eppinger SD (2013) The guide to the product management and marketing body of knowledge: ProdBOK(R) Guide, 1st edn. Product Management Educational Institute

Ghazi AN, Petersen K, Reddy SSVR, Nekkanti H (2019) Survey research in software engineering: problems and mitigation strategies. IEEE Access 7:24703–24718. https://doi.org/10.1109/ACCESS.2018.2881041

Goldkuhl G (2011) Pragmatism vs interpretivism in qualitative information systems research. Eur J Inf Syst 21:135–146. https://doi.org/10.1057/EJIS.2011.54

Harpe SE (2015) How to analyze Likert and other rating scale data. Curr Pharm Teach Learn 7:836–850. https://doi.org/10.1016/j.cptl.2015.08.001

Hove SE, Anda B (2005) Experiences from conducting semi-structured interviews in empirical software engineering research. In: Proc – Int Softw Metrics Symp, pp. 203–212. https://doi.org/10.1109/METRICS.2005.24

International Institute of Business Analysis (2015) A guide to the business analysis body of Knowledge® (BABOK® Guide) v3, IIBA

ISO/IEC 31010:2019 (2019) Risk Management – Risk Assessment Techniques. International Organization for Standardization

ISPMA SPM Framework V.1.3 (n.d.). https://ispma.org/ispma-spm-framework-v-1-3/

Jantunen S, Hietaranta K, Gause DC (2013) Adjusting to increasing product management problems: Challenges and improvement proposals in one software company. In: Doerr J, Opdahl AL (eds) Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). Springer Verlag, pp 386–400. https://doi.org/10.1007/978-3-642-37422-7_28

Kagan P, Naumova A, Vilman Y (2016) The Problems of project management software implementation in construction corporations. In: MATEC Web Conf. 73, TPACEE-2016, EDP Sciences. https://doi.org/10.1051/matecconf/20167307016

Kitchenham BA (2004) Procedures for Undertaking Systematic Reviews, Computer Science Department, Keele University (TR/SE-0401) and National ICT Australia Ltd. (0400011T.1)

Kitchenham BA, Pfleeger SL (2008) Personal opinion surveys. In: Shull F, Singer J, Sjøberg DIK (eds) Guid To Adv Empir Softw Eng. Springer, London. https://doi.org/10.1007/978-1-84800-044-5_3

Kittlaus H-B (2012) Software product management and agile software development: conflicts and solutions. In: Maedche A, Botzenhardt A, Neer L (eds) Softw People Manag Prof. Springer, Berlin, pp 83–96. https://doi.org/10.1007/978-3-642-31371-4_5

Kittlaus H-B, Fricker SA (2017) Software product management: The ISPMA-compliant study guide and handbook, Springer. https://doi.org/10.1007/978-3-642-55140-6

Kristinsdottir S, Larusdottir M, Cajander Å (2016) Responsibilities and challenges of product owners at Spotify – an exploratory case study, Volume 9856 LNCS, 2016, Pages 3–16, IFIP WG 13.2/13.5 Joint 6th International Conference on Human-Centered Software Engineering, HCSE 2016 and 8th International Conference on Human Error, Safety, and System Development, HESSD 2016; Stockholm; Sweden; 29 August 2016 through 31 August 2016; Code 180489

Lopez J, Juárez-Ramírez R, Huertas C, Jiménez S, Guerra-García C (2016) Problems in the adoption of agile-scrum methodologies: A systematic literature review. In: pp. 141–148. https://doi.org/10.1109/CONISOFT.2016.30

Maglyas A, Nikula U, Smolander K (2012a) Lean solutions to software product management problems. IEEE Softw 29:40–46. https://doi.org/10.1109/MS.2012.108

Maglyas A, Nikula U, Smolander K (2012b) Comparison of Software Product Management Practices in SMEs and Large Enterprises. Lect Notes Bus Inf Process 114:15–26. https://doi.org/10.1007/978-3-642-30746-1_2

Maglyas A, Nikula U, Smolander K (2012c) "What do practitioners mean when they talk about product management? 20th IEEE International Requirements Engineering Conference (RE), pp. 261–266. https://doi.org/10.1109/RE.2012.6345812

Maglyas A, Nikula U, Smolander K (2013) What are the roles of software product managers? An empirical investigation. J Syst Softw 86:3071–3090. https://doi.org/10.1016/j.jss.2013.07.045

Maglyas A, Nikula U, Smolander K, Fricker SA (2017) Core software product management activities. J Adv Manag Res 14:23–45. https://doi.org/10.1108/JAMR-03-2016-0022

Murphy-Hill E, Zimmermann T, Bird C, Nagappan N (2015) The design space of bug fixes and how developers navigate it. IEEE Trans Softw Eng 41:65–81. https://doi.org/10.1109/TSE.2014.2357438

Palinkas LA, Horwitz SM, Green CA, Wisdom JP, Duan N, Hoagwood K (2015) Purposeful sampling for qualitative data collection and analysis in mixed method implementation research. Adm Policy Ment Heal Ment Heal Serv Res 42:533–544. https://doi.org/10.1007/s10488-013-0528-y

Pragmatic Institute (2020) The pragmatic Framework. https://www.pragmaticinstitute.com/framework/

Project Management Institute (PMI) (2017) A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 6th ed., PMI

Punter T, Ciolkowski M, Freimut B, John I (2003) Conducting on-line surveys in software engineering, Proc. – 2003 Int Symp Empir Softw Eng. ISESE 2003. 80–88. https://doi.org/10.1109/ISESE.2003.1237967

Scaled Agile Inc. (2019) White Paper: Achieving Business Agility with SAFe ® 5.0. https://www.scaledagile.com/resources/safe-whitepaper/

Schwaber K, Sutherland J (2017) The scrum guide™ the definitive guide to scrum: the rules of the game. Scrum.org

Shahin M, Zahedi M, Babar MA, Zhu L (2019) An empirical study of architecting for continuous delivery and deployment. Empir Softw Eng 24:1061–1108. https://doi.org/10.1007/s10664-018-9651-4

Sommerville I (2015) Software engineering, 10th edn. Pearson

Springer O, Miler J (2018) The role of a software product manager in various business environments. Proc 2018 Fed Conf Comput Sci Inf Syst FedCSIS 2018, Institute of Electrical and Electronics Engineers Inc., pp. 985–994. https://doi.org/10.15439/2018F100

The Stationery Office (2010) Management of Risk: Guidance for Practitioners, 3rd edn. Office of Government Commerce

Uusitalo P, Seppänen O, Peltokorpi A, Olivieri H (2019) Solving design management problems using lean design management: the role of trust. Eng Constr Archit Manag 26:1387–1405. https://doi.org/10.1108/ECAM-03-2018-0135

Wagenblatt T (n.d.) Management for Professionals Software Product Management. http://www.springer.com/series/10101

Walsham G (2006) Doing interpretive research. Eur J Inf Syst 15:320–330. https://doi.org/10.1057/PALGRAVE.EJIS.3000589

Wohlin C (2014) Guidelines for snowballing in systematic literature studies and a replication in software engineering, ACM Int. Conf. Proceeding Ser. https://doi.org/10.1145/2601248.2601268

Zhou X, Jin Y, Zhang H, Li S, Huang X (2016) A map of threats to validity of systematic literature reviews in software engineering, Proc – Asia-Pacific Softw Eng Conf APSEC 0: 153–160. https://doi.org/10.1109/APSEC.2016.031