

RESEARCH ARTICLE

Strategies for Dealing With Software Product Management Challenges

OLGA SPRINGER¹, JAKUB MILER¹, AND MICHAŁ R. WRÓBEL¹

Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 80-233 Gdańsk, Poland

Corresponding author: Olga Springer (olga.springer@pg.edu.pl)

ABSTRACT Software product management, which involves multiple processes and responsibilities and links to many activities within the organisation, increases the success rate of IT projects. However, the adoption of product management activities and the implementation of the software product manager role itself varies between companies. This research explores the topic of software product managers' strategies for dealing with challenges in their work. The aim of the study is to identify and evaluate solutions to the main problems that affect the software product management process and to provide guidelines for dealing with them. For this purpose, a focus group method was adopted and 47 software product managers participated in 15 focus groups. This was followed by a survey to assess the comprehensibility and effectiveness of the solutions identified. The paper proposes a list of 39 solutions to the 5 most common problems in the work of product managers. Furthermore, based on the identified solutions and the results of discussions with experienced professionals during the focus groups, guidelines for these problems were also developed. These guidelines can be used by product managers as well as other roles working in the product development team to create their own strategies for overcoming software product management challenges and improving software engineering practices. Finally, the solutions and guidelines presented, when combined with a list of problems identified in previous research, will form a Software Product Management Guide – a framework currently under construction, as part of wider research, for product development teams to improve software engineering practices.

INDEX TERMS Product management, product manager, software product management.

I. INTRODUCTION

In 2007 it was proved that systematic product management increases the success rate of software projects [1]. Procter and Gamble is attributed as the first company that utilised product management as an explicit management process. [2]. The characteristics of the product management in FMCG was that it was more focused on marketing aspects – like brand, communication, pricing, promotions, rather than product development [3].

Later on, when new industries (including technology) emerged, organisations could not just focus and rely on brand management and marketing, because other areas started significantly impacting product success (e.g. technology).

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Ali Babar¹.

This brought product development back to the centre of the product management focus. Right now marketing is more responsible for the brand and customer acquisition, while the product team's goal is to define the right value proposition and to develop the software products [3].

Product management can be defined in several ways. According to Heines product management is the business management of products, product lines or product portfolios, in a comprehensive manner, to maximise value throughout their life cycle [4]. Kittlaus et al. provide a definition from a different perspective. According to them, product management is the combination of goods and services that a supplier/development organisation assembles to support its commercial interests in order to transfer certain rights to the customer [2]. Gorchels' definition of a product manager seems the closest to our understanding. The software

product manager is responsible for what the product is, how it works, whom it serves and how it affects the company and customers [5].

The idea of product management also fits well with the agile approach to IT project management, where one of the main principles is the high availability of a representative of the customers and clients of the product being produced. Scrum Guide introduced the role of the Product Owner, which is responsible for business goals and value [6]. The problem is that, according to the Scrum definition, the product backlog contains everything that is related to the product, and therefore by inference the Product Owner becomes responsible for everything related to the product. This is a formula that doesn't scale to accommodate companies with complex products or large software development projects [7]. Defining the Product Owner role may be challenging in the organisation [8]. The Product Owner role in software companies that develop their own products is very often held by product managers as they have the proper competencies required to lead and achieve business goals [9].

Software product management covers many processes and responsibilities and connects to many activities carried out in the company. The adoption of product management activities varies between companies [10]. An attempt to adopt all activities instantly has been identified as a common problem in implementing product management. Companies face common problems in the adoption of software product management, like long release cycles, short-term thinking, and a lack of constant collaboration between organizations and customers [11]. Successful product management means delivering the right products at the right time for the right markets [12]. Additionally, software product management is recognised as the function responsible for all aspects of a software product, during the whole lifecycle of this product, from its conception to the end of its life [13].

Product managers work between business, technology and user experience. They know the market, competitors and target group. They use that knowledge to develop solutions that are beneficial for the company and provide value for customers [14], [15]. Product managers provide the expertise needed to lead and make strategic product decisions. The role of product manager spans many activities from strategic to tactical and provides important cross-functional leadership – between management, engineering, marketing, sales and customer success teams [2].

The complexity of this role makes it very demanding. There are many problems that affect the work of a software product manager. These problems reduce software product management effectiveness and make it more difficult to achieve business goals ultimately [16].

We recognised an increased demand for product managers in IT organisations [17], [18]. Growing popularity and challenges of the role have not yet been investigated deeply and we believe it is gaining more significant impact on the organisations each year. The recent bibliographic study that

reviewed evolution of Software Product Management pointed out, that surprisingly the software product management has been addressed very little in the science. Software product management as a field seems to be fragmented and would benefit from setting up an ambitious research agenda to guide the development of the field [19].

The contribution presented in this paper is the result of the second phase of multi-phase research project aimed at building a framework to support product management activities in IT companies: Software Product Management Guide.

Software Product Management Guide is a framework that allows teams to evaluate their key challenges and presents them with how experienced Software Product Managers solve the problems related to software product management.

Software Product Management Guide consists of the following: - 27 problems with perceived frequency and severity. - Guidelines and solutions for the top 5 frequent problems. - Techniques that SPMs use while working on the specific problems

The first research phase to prepare a framework covered the literature review to confirm the research gap and set the research scope and context. The next step was the identification of problems faced by software product managers in their work, and the evaluation of perceived frequency and perceived severity of the selected common problems [16].

This paper presents the next phase focusing on solutions to the most common problems in the work of software product managers.

The aim of this research is to look for potential solutions to recognised problems and build a set of recommendations on applying these solutions in different business contexts. As far as we know there is no comprehensive method available in the current literature to solve problems related to software product management and impact the performance of this process ultimately. We believe, as part of the wider research we can provide a framework to help product managers work effectively and increase the business impact.

The main goal of this research is to **identify and evaluate the solutions to the most relevant software product management problems** from our previous paper [16]. For this purpose, three research questions were posed:

- RQ1.** How do product managers solve the identified problems in their work?
- RQ2.** To what extent reported solutions to the most common problems are comprehensible?
- RQ3.** What is the effectiveness of these solutions in solving the problem?

The paper is structured as follows. Section II provides some related work. In Section III we describe research method including a focus group and survey. Section IV presents identified strategies for the most common software product management problems in the form of guidelines and solutions. Finally, Section V presents results discussions and Section VI final conclusions.

Software Product Management Guide

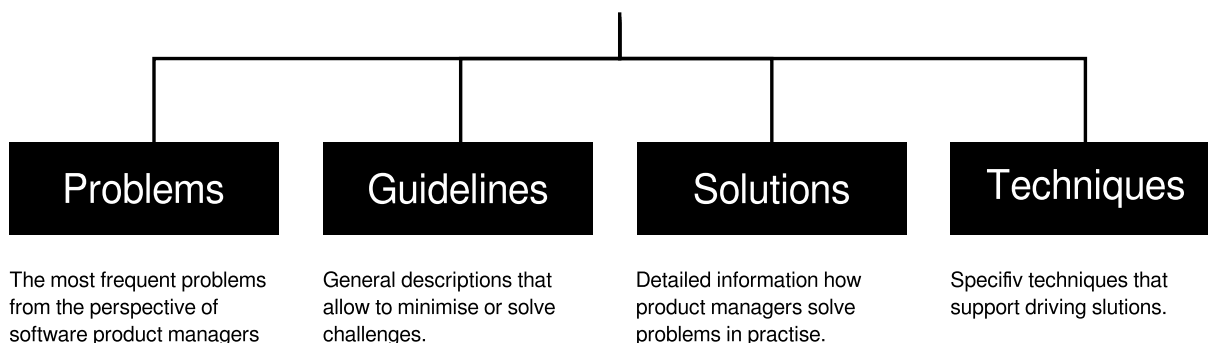


FIGURE 1. Software product management guide.

II. RELATED WORK

In previous research [16] we made a systematic literature review to identify the most important sources explaining software product management problems from the perspective of software product managers. The data were collected by a systematic literature review (5 main databases analyzed), interviews (10 software product managers from IT companies), and survey (89 participants).

First, we analysed the data from 5 databases. Challenges related to unclear responsibilities and strategy and the difficulty of the decision-making process were mentioned the most in scientific publications. Additionally, we conducted our own research where we interviewed and surveyed practitioners about the challenges they face in their daily work. In total, 95 problems have been identified during 10 interviews, which have been narrowed down to 27 problems based on their occurrence in at least 3 interviews. These selected problems were further prioritized by perceived frequency and severity based on the questionnaire survey among software 89 product managers [16].

The analysis revealed that the data received on the most severe and frequent problems is consistent with the results of the literature review, but it also identified other areas where problems occur and impact software product managers. Some of the identified problems in our research spanned beyond the software product management process itself, but they all affect the work of software product managers [16].

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

- **P74.** Determining the true value of the product that the customer needs.
Product managers have to run research and work iteratively with teams to understand the customer needs, scaling opportunities and customer willingness to pay for the product.
- **P35.** Strategy and priorities are changing frequently.
One of the organisational problems is affecting Product Managers and other roles widely. When the strategy is changing frequently, Product Managers and their teams struggle with prioritisation, don't see the long-term

picture and can't achieve outcomes, as the direction is changing too often.

- **P9.** Technical debt.
The challenge that is slowing down the product development process and making it even more difficult to prioritise the Product Roadmap. Technical debt is hard to measure but needs to be watched and prioritised by teams wisely. Good cooperation between Product&Engineering is required to either minimise, pay it off completely or accept the risk.
- **P64.** Working in silos (problem with communication, synchronisation between teams).
Initiatives run across different departments require more discipline to align teams around common goals, synchronise them and organise communication that will rather support teams than slow them down.
- **P69.** Balancing between reactive and proactive work.
For mature products, it is always a challenge to prioritise research, innovation and new value for customers against a queue of bugs, escalation, maintenance-related work.

The identified problems show that Software Product Managers also struggle with gathering data, requirements management, and accessing customers in order to develop valuable solutions. Some of the problems are specific to the Software Product Manager role, but a significant part of them are organisational challenges that impact other roles as well [16].

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

A. THE IMPACTS OF SOFTWARE PRODUCT MANAGEMENT

Ebert's study from 2007 showed that time to market, schedule adherence and handover quality improves with the strengthening of a coherent product management role [1]. Explanatory factors indicated as having a positive impact on product management have been explained and coined into guidelines towards successful product management presented in Table 1.

TABLE 1. Factors with positive impact on product management [1].

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

B. LEAN SOLUTIONS TO SOFTWARE PRODUCT MANAGEMENT PROBLEMS

In 2012, Maglyas published a guide on how to apply lean practices to solve software product management problems. In the research, they studied 13 organizations to get an understanding of how software product management practices were adopted [20].

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

Table 2 shows software product management problems and solutions identified by Mayglayas et al. [20].

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

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

C. MANAGING SOFTWARE PRODUCTS IN A GLOBAL CONTEXT

In 2018 Christof Ebert published another article [21] that provides concrete practices to fertilize and evolve software

product management in global teams. 75 recipients participated in the survey on global software product management. Additionally, the authors interviewed twenty representative companies. Results mention three product management-related problems, however, no solutions are proposed for software product managers to eliminate them:

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

Other findings from the research:

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

D. PRODUCT YIELD POTENTIAL RADAR (PYPR)

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

TABLE 2. Product management problems and solutions [20].

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

product success [22]. It describes 6 dimensions and 31 success factors that determine continuous and sustainable product success. The methodology illustrates the dimensions and success factors that product managers, product teams, and product organizations need to have on their radar.

The framework explains how to leverage and adapt the software product management with regard to aspects like product viability, product development, product marketing and software demonstrations and training, as well as more general aspects like markets, customers and organizational maturity.

PYPR is a framework that enables product managers, product teams and organisations to balance what really matters in product management. The approach underlines it is essential for product managers to understand the good and weak areas of the product in order to become excellent and achieve product success.

The PYPR framework consists of four steps:

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

Product Yield Potential Radar PYPR is comprehensive knowledge about the product management process. The book mentions and describes many practices and techniques that can be used to optimise it.

E. IDENTIFIED RESEARCH GAP

None of the previous studies that have been identified has focused strictly on providing solutions that product managers can use to solve the problems they encounter in their work. Rather, they explain the role, and analyse the challenges, but only list high-level guidelines for product managers that are difficult to implement in practice.

We address this gap and provide insights into the topic – for the most common challenges we recognised in the previous research [16], we have identified how product managers solve them in practice. Based on the data collected, we have developed detailed guidelines on how to deal with the top frequent problems in a product manager's work.

III. STUDY DESIGN

The aim of this paper is to identify and evaluate a set of solutions to the most important problems that arise in the work of product managers. A mix-method approach was used for this purpose. We have applied the sequential exploratory strategy [23], which additionally allowed for increased accuracy and reliability with data triangulation. To answer RQ1, i.e. to identify solutions used by practitioners in their daily work, the focus group method was applied, as it is reported to be suitable for gathering experience from expert groups [24]. Whereas a questionnaire survey was used to evaluate and select the most comprehensible (i.e. clear and easy to understand) and effective of the proposed solutions (RQ2, RQ3). This is consistent with Kontio et al. guidelines, which indicate that focus groups are typically not the only research method used in a study [24].

A. FOCUS GROUPS STUDY

The survey using the focus group method was conducted in accordance with the guidelines proposed by Kontio et al. [24], [25], which were developed on the basis of feedback and experiences from software engineering practitioners. These guidelines have been adopted as a common approach to conducting research using this method in the field of software engineering.

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

1) PLANNING THE RESEARCH

The purpose of the focus group study was to identify solutions to common problems that product managers face in their daily work. In the previous study, we have recognized 27 such problems [16]. In order to conduct a productive study, it was decided to divide the problems into categories, where each one would be the subject of separate focus groups. We used a posteriori classification, assigning 2 to 4 problems to one of 8 categories.

Table 3 shows the list of categories with descriptions and assigned problems. For consistency, problems are numbered as in the source article [16].

2) DESIGNING FOCUS GROUPS

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

In order to identify and engage such experienced professionals, we announced an initiative called Product Management Challenge. A webpage with the description of the planned research was prepared¹ and distributed in social media groups where software product managers participate. It was also shared by the authors of this article directly through their LinkedIn profiles.

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

The initiative was promoted under the idea of building a community of experienced software product managers. As a

benefit of joining, participants would have the opportunity to share common experiences during focus group sessions.

In order to collect data on the software product management experience in our sample, we asked the participants of the Product Management Challenge to fill in a questionnaire. We asked the following questions:

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

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

While revising the solutions to these 5 selected problems, we noticed that there were still several duplicated solutions to the same problem from two separate focus groups organised for the same category. Therefore, we carried out an additional merging phase between the groups. The duplicates were merged in 2 steps:

- individual analysis of the content of the solutions by both authors of the article where pairs of solutions proposed for merging were marked,
- meeting and discussion of the proposed pairs until a decision was reached whether or not to merge a particular pair of solutions.

The results of the merging are shown in Table 4.

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

For each solution, respondents were asked to rate in a Likert 5-point response format (strongly disagree, disagree, neutral, agree, strongly agree) [39] the accuracy of the following two statements:

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

We defined comprehensibility of the problem solution as the total percentage of "agree" and "strongly agree" responses to the first question, and perceived efficiency as the analogous percentage of responses to the second question. It is important to notice that this definition is based on

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

TABLE 3. List of categories and assigned problems [16].

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

TABLE 4. Second phase of solution merging.

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

respondents' opinions, which is typical to qualitative research such as ours.

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

The initial goal was to get at least 17 responses (response rate greater than 30%) for each questionnaire. In order to

achieve a satisfying response rate, besides e-mail communication experts were also contacted individually by authors through LinkedIn with requests to complete at least a few surveys they had been invited to.

Table 5 shows the number of respondents who took part in evaluating solutions to each problem.

Based on achieved results from the research, as well as discussions during the focus groups, authors cooperated on defining guidelines for the top frequent problems related to software product management. The authors analysed the solutions for a particular problem, reviewed the notes from

focus groups, and discussed key learnings. The leading author formed the guideline and asked other researchers for revision. Guidelines represent high-level insights on how to tackle specific problems.

IV. STUDY RESULTS

Two acceptance thresholds were established for the evaluated solutions, 80% for comprehensibility and 60% for efficiency. Additionally, solutions with a number of negative ratings above 20% (strongly disagree, disagree) were excluded. This resulted in the rejection of 43 out of all 82 solutions.

In order to focus on the high quality of results, we set up a higher expectation for the comprehensibility factor. The goal was to exclude the results that are hard to read and difficult to interpret.

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

Table 6 shows details of the number of accepted solutions for each problem along with the number of solutions rejected based on respective exclusion criteria.

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

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

A. PROBLEM: DETERMINING THE TRUE VALUE OF THE PRODUCT THAT THE CUSTOMER NEEDS (P74)

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

Guidelines for this problem:

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

To solve this problem, we propose the following 7 solutions.

1) SOLUTION 1: CUSTOMER FEEDBACK LOOP

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

- 1) Integration of many communication channels with the client – gathering all customer feedback in one tool, e.g. ProductBoard (chats with clients, Intercom, reports from sales meetings). The Slack channel is where the Customer Service Team can submit issues/feedback and everyone can share their product ideas.
- 2) Automate as much as possible. The option to report feedback should be available in the tools other employees use on daily basis. These channels should be integrated with the feedback management tool.
- 3) Weekly review of new tickets that appeared in the feedback management tool.
- 4) Classification of reported ideas – in terms of appropriate tagging or connection with another ticket that already exists.
- 5) Each ticket, when tagged, is additionally marked with priority and complexity estimation.
- 6) The tool itself builds a hierarchy of the most frequently reported problems and needs.

2) SOLUTION 2: USER TESTING SESSIONS – TESTING PROTOTYPES

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

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

3) SOLUTION 3: IN-DEPTH INTERVIEWS WITH POTENTIAL/FUTURE USERS OF THE PRODUCT

Comprehensibility: 100%, Effectiveness: 85,19%

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



TABLE 5. Participation in the questionnaire survey.

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

TABLE 6. Accepted solutions after the questionnaire survey.

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

4) SOLUTION 4: DATA ANALYSIS AND IN-DEPTH INTERVIEWS WITH CLIENTS

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

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

5) SOLUTION 5: VALIDATION OF WHETHER WE ARE SOLVING THE CUSTOMER’S PROBLEM BASED ON SIMPLE PROOF OF CONCEPT

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

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

6) SOLUTION 6: JOBS TO BE DONE

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

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

7) SOLUTION 7: VALUE PROPOSITION CANVAS

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

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

B. PROBLEM: STRATEGY AND PRIORITIES ARE CHANGING FREQUENTLY (P35)

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

Guidelines for this problem:

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

To solve this problem, we propose the following 6 solutions.

1) SOLUTION 1: STRATEGY AND PRODUCT-CENTRED PROCESS

Understability: 100%, Effectiveness: 96,55%

- 1) Cooperate with the stakeholders on the product vision.
- 2) Together with the stakeholders elaborate on the main business goal/ business challenge to achieve on the market to achieve the state described in the product vision.
- 3) Together with the team divide the goal/challenge into smaller measurable goals to achieve in the product to get closer to the main goal.
- 4) Clearly describe the current state of the product.
- 5) Aim to employ and build a bigger product team.
- 6) Develop the product-centred process starting from problem collection up to solution development including prioritization, transparency and roadmap.
- 7) Promote the process and strategy.

2) SOLUTION 2: YIELD POTENTIAL (INCREMENTAL)/COST OF DELAY + URGENCY PROFILE

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

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

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

3) SOLUTION 3: TAKING INTO ACCOUNT THE ONBOARDING OF NEW CUSTOMERS, INTEGRATIONS, AND CUSTOM-TAILORED DEPLOYMENTS WHEN PLANNING

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

This solution addresses the problem when the onboarding of new big customers was not included in the yearly plans and finally, it won over other highly prioritised tasks.

With B2B products, we often face Enterprise customers that use custom-tailored solutions, thus the increased onboarding effort might win over other tasks.

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

4) SOLUTION 4: "NOW, NEXT, LATER" ROADMAP

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

- 1) Assuming the product is in the growth phase and the changes to the strategy are justified, frequent strategy changes should not be the problem – we are searching for new product solutions, and new ways to win on the market. We should not fight it, we should embrace it.
- 2) The solution is to develop "soft roadmaps". In case of frequent changes arrange with the management to define the roadmap in "Now, Next, Later" blocks instead of time. Show uncertainty on the roadmap.
- 3) This roadmap makes the product more flexible and adaptable to the changing environment.

5) SOLUTION 5: DEFINE STRATEGY

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

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

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

C. PROBLEM: TECHNICAL DEBTS (P9)

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

Guidelines for this problem:

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

To solve this problem, we propose the following 9 solutions.

1) SOLUTION 1: SMUGGLING

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

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

2) SOLUTION 2: MEASURE AND BUILD AWARENESS

Comprehensibility: 100%, Effectiveness: 71,43%

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

- 1) Measure your debt. Few options:
 - Using ready-made tools, like Sonarqube shows how much technical debt the code contains.
 - Using can use one of the following measures, for example, division of maintenance and development tasks, the number of bugs reported, and application performance. It is worth looking for good examples of measuring debt on the market and choosing a few of the most important.
 - Setting up an Epik in Jira which includes all technical-debt-related issues.
- 2) Building debt awareness in the organization, educating what we are dealing with.

3) SOLUTION 3: A CONSTANT TOPIC AT THE SPRINT REVIEW MEETING

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

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

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

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

4) SOLUTION 4: PRODUCT OWNER TAKING THE INITIATIVE – A SIGNAL TO THE TEAM THAT IT IS WORTH TAKING CARE OF THE TECHNICAL DEBT

Comprehensibility: 100%, Effectiveness: 90,48%

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

5) SOLUTION 5: PRODUCT ROADMAP – ANALYSIS AND UPDATE

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

- 1) Analysing technical debt by the team and defining areas to be addressed.
- 2) Estimating work required to reduce technical debt in specific areas.
- 3) Establishing priorities to be addressed taking into account the estimate and impact on the product (technical and business).
- 4) In the case of larger topics, adding initiatives to the Roadmap related to the reduction of technical debt. While working on a given area in connection with another ongoing initiative, we make an overhead to also deal with the technical debt.

6) SOLUTION 6: MANAGING TECHNICAL DEBT

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

- 1) When a project is starting from scratch – agree with the team and technical leaders that any noticed or conscious shortcomings are reported and aggregated within, for example, one epic. With an inherited product – analysis of the existing state and preparation of an initial optimization list.
- 2) Educating customers – why it is so important to identify technical debt (impact on the product/business/users) and what value will the optimisations (i.e. refactoring, library uplift) bring ultimately. Build awareness of debt as something normal in the software development process that needs to be managed. Share examples (i.e. LinkedIn rewritten the entire core from Ruby to NodeJS because at some point development/maintenance/scaling was too cumbersome/expensive).
- 3) Implementation of the rule – leave the code better than you found it.
- 4) Reservation of time for regular debt reduction as part of the sprint / including the necessary refactors / corrections in the task estimates.

7) SOLUTION 7: TAKING INITIATIVE AND PROPOSING CHANGE

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

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

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

8) SOLUTION 8: PRIORITISATION + COMMUNICATION TRANSPARENCY + OWNERSHIP

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

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

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

9) SOLUTION 9: CONSTANT % OF TIME SPENT ON TECHNICAL DEBT

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

- 1) Analysing the volume of technological debt (in the backlog) – i.e. external company audit in the field of system security, or internal analysis.
- 2) Ongoing backlog updates with tasks related to technological debt. Additionally, once a quarter / half a year, brainstorm to define new things / analyze the size of technical debt.
- 3) Estimating work for every individual element of technical debt and prioritisation.
- 4) In each sprint, allocate a certain amount of time (eg 10%) to tasks related to technological debt. Picking up technical debt tickets for each sprint (depending on severity and criticality).

10) SOLUTION 10: THE BOY SCOUT RULE - CLEAN UP

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

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

D. WORKING IN SILOS (PROBLEM WITH COMMUNICATION, SYNCHRONISATION BETWEEN TEAMS) (P64)

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

Guidelines for this problem:

- 1) Define strategic product goals to which particular teams would contribute.

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

To solve this problem, we propose the following 11 solutions.

1) SOLUTION 1: PRODUCT TEAM DEMOS

Comprehensibility: 100%, Effectiveness: 71,43%

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

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

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

2) SOLUTION 2: DEVELOP CLEAR RULES FOR COLLABORATION, DEFINE PROCESSES AND RESPONSIBILITIES, PROMOTE TEAMWORK

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

- 1) Involve the team members in making key decisions e.g. on architecture.
- 2) Define the process that is comprehensible, transparent and easy to apply.
- 3) Apply a single source of truth in the context of both requirements and organizational culture.
- 4) Promote teamwork, even in code programming, and ensure the team sees the value in teamwork.
- 5) Carry out the daily meetings correctly – moderated, including status, goals and problems.
- 6) Coordination between teams.

3) SOLUTION 3: STRATEGIC GOALS

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

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

4) SOLUTION 4: INCREASE TEAM AUTONOMY

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

- 1) As a Product Owner build the competencies and ensure sharing of knowledge. Find other teams that have the competencies needed and invite more experienced people to less experienced teams. Build an autonomous team with all the necessary competencies.

- 2) “Review” meeting to analyze the implemented solution from the development, customer and UX point of view.

5) SOLUTION 5: INTRODUCE PROCESS TO SYNCHRONIZE THE WORK OF DIFFERENT PRODUCT TEAMS

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

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

Examples:

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

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

6) SOLUTION 6: CREATE MULTIDISCIPLINARY TEAMS FOR INITIATIVES

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

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

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

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

7) SOLUTION 7: REGULAR MEETINGS OF POs AND PROJECT MANAGERS

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

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

8) SOLUTION 8: WORKING TOGETHER ON THE ROADMAP BY ALL TEAMS INVOLVED IN PRODUCT DEVELOPMENT

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

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

9) SOLUTION 9: KICK-OFF MEETING BEFORE PROJECT START

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

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

10) SOLUTION 10: TRANSPARENT AND PERIODIC COMMUNICATION

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

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

E. PROBLEM: BALANCING BETWEEN REACTIVE AND PROACTIVE WORK (P69)

To make sure teams spend enough time on proactive work, Product Managers recommend finding a way to guarantee the time for proactive initiatives at the level of team goals. They also believe the time is required to run research in order to create a product vision. In case when it is not possible to involve all teams in proactive initiatives, it is recommended to create a multidisciplinary team dedicated to running product discovery.

Guidelines for this problem:

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

To solve this problem, we propose the following 8 solutions.

1) SOLUTION 1: SETTING GOALS FOR PRODUCT DEVELOPMENT – DIVIDING THE TIME TO DEVOTE TO REACTIVE AND PROACTIVE WORK

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

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

- 1) Defining OKRs / OITs based on the product strategy.
- 2) Marking the elements of the Product Backlog that are connected with OKRs and those that do not relate to them.
- 3) In the Sprint Backlog, 70% of tasks directly contribute to strategies / OKRs, 30% to bugs or other more ad hoc ideas.
- 4) Measuring real team effort devoted to the development of new functionalities and maintenance.

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

2) SOLUTION 2: FORECASTING

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

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

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

3) SOLUTION 3: CONTINUOUS PRIORITIZATION (E.G. BY RICE METHOD)

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

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

For prioritization, using one of the methods, i.e. the RICE to evaluate items based on four variables:

- 1) Reach
- 2) Impact
- 3) Confidence
- 4) Effort

4) SOLUTION 4: INTRODUCE PRODUCT DISCOVERY PROCESS IN THE ORGANISATION

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

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

5) SOLUTION 5: CUSTOMER ADVISORY BOARD

Comprehensibility: 100%, Effectiveness: 64,71%

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

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

6) SOLUTION 6: NORTH STAR

Comprehensibility: 100%, Effectiveness: 82,35%

- 1) Setting product development goals / supporting innovation (Roadmap, OKR) (prioritization).
- 2) Keeping teams/work focused on the goals defined. Work synchronisation between teams.
- 3) Close cooperation with clients to define their long-term needs (input to the roadmap).
- 4) Cooperation with internal stakeholders.
- 5) Visualization of the goal with mockups.
- 6) Defining the speed of teams and defining the commitment to work for current clients (dev/bugfix).
- 7) Monitoring of application and user behaviour.
- 8) Removal of unused functionalities from the product (simplification of maintenance and further development)

7) SOLUTION 7: DEFINE GOALS (OKRs) AND PRODUCT ROADMAP

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

- 1) Defining high-level goals (OKRs – Objective and Key Results) and going down with them to the teams. The designation of OKRs for the “team” included both discovery and delivery objectives to balance the ratio of proactive and reactive work.
- 2) Defining the product roadmap.
- 3) Setting sprint goals in line with the goals and roadmap.
- 4) Planning of work in accordance with the purpose of the sprint.
- 5) Appointing a person per sprint responsible for addressing unplanned tasks (eg defects) or leaving a buffer in the team's ‘capacity’ for addressing these tasks.
- 6) Building assertiveness in the development team.
- 7) Saying ‘NO’ to tasks inconsistent with the goals – using appropriate argumentation.

8) SOLUTION 8: DEDICATED TEAM TO MAKE A PRODUCT DISCOVERY

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

Creating a dedicated, multidisciplinary team for Product Discovery (consisting of various competencies: eg UX Researcher, Marketing Manager, Product Manager).

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

- 1) Focusing on analyzing and exploring the problem that we solve for the client and looking for insights that can be used in development.
- 2) A constant R&D process in which we conduct research, identify opportunities and test potential solutions. Creating a list of opportunities, potential solutions, and

experiment backlog. Management and summary of experiments (success criteria, conclusions).

- 3) Agile/iterative definition of what we should build now to deliver value to the user in such a way as to deliver business value.
- 4) Testing/hacking, looking for solutions 10x better rooted in the company's culture.

9) SOLUTION 9: IMPLEMENTATION OF MAINTENANCE AND CUSTOMER SERVICE MANAGEMENT STANDARDS

Comprehensibility: 100%, Effectiveness: 76,47%

- 1) Analysis of the causes of incidents.
- 2) Learning from incidents in order to minimize their occurrence in the future.
- 3) Support and optimization of product-related processes to minimize the need for reactive work.

10) SOLUTION 10: DEFINE THE PROBLEM, PRIORITISE AND CREATE A ROADMAP

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

The strategic team consist of Product Manager, Designer, Analyst, and Engineering Manager) – responsible for research, and benchmarking. A separate backlog is being managed, focused on Discovery related topics.

- 1) Problem definition:
 - a) Collection of facts (feedback from users, internal and external stakeholders; benchmarking, tracking data, IDI with users) – cooperation with Product Designer, Product Analyst, Researchers, EM.
 - b) Putting hypotheses to be tested, planning experiments.
 - c) Conducting experiments and confirming the problems and proposed solutions.
- 2) Prioritization:
 - a) Collecting problems with proposed solutions.
 - b) Workshop using MOSCOW method – gathering all stakeholders and a person who will be the user's advocate, prioritizing in terms of what must, should, could, would be done, from the perspective of the user, business and other important perspectives.
 - c) Arranging prioritized solutions on a solution pyramid to answer the question of where (base, middle, tip) these solutions rank for the problems confirmed in the experiments.
 - d) One Roadmap – setting up a strategy to deliver in the shortest possible time the greatest value for the user, business, and all others. Setting up precise goals.

V. DISCUSSIONS

The main contribution of this article is the list of 39 descriptive solutions for the top 5 frequent problems software product managers face in their work. This list is derived from the results of research conducted using focus groups to identify

solutions and questionnaire surveys to evaluate their effectiveness and comprehensibility.

Altogether, 15 focus groups were organised between December 2020 and March 2021 to analyse the solutions for the top frequent 27 problems related to software product managers. The authors decided to continue the research narrowing the scope down to the 5 top frequent problems, as the required amount of work and participants' engagement to research the whole originally planned scope, was not possible to achieve. The choice of 5 problems was a reasonable trade-off between the workload and value brought to the field of product management. A total of 81 unique solutions were identified for these 5 top frequent problems, which answer the RQ1 *How do product managers solve the identified problems in their work?*.

Additionally, a set of questionnaire surveys was conducted which resulted in an evaluation of the comprehensibility (RQ2) and effectiveness (RQ3) of the identified solutions. Based on the evaluation, 39 solutions were accepted. Additionally, the description of these solutions was enriched with concise guidelines on how to deal with each problem.

A. COMPARISON WITH PROBLEMS FROM THE PREVIOUS RESEARCH

Our findings are compatible with previous works in terms of the high-level guidelines for software product managers.

C. Ebert [1] identified 4 general guidelines for product managers related to business objectives and accountability, mastering requirements, managing risks and uncertainty, leadership and teamwork. The guidelines and solutions presented in this paper cover these topics as well, but are designed for the specific problem of software product management and that makes them incomparable.

The solutions identified by Maglayas [20] are designed for 5 specific problems, however, these are rather high-level guidelines than specific solutions to be applied in the organisation. Our research proposes both guidelines as well as a set of descriptive solutions which may be used in a different environment.

Our research allowed identify strategies for additional problematic areas that weren't analysed in previous research. There are many problems that may have an impact on a Software Product Manager's job, and they can be related to their core activities or lie beyond them. An example of a problem that in reality very often needs the software product manager's attention is technical debt (P9) – which is the 3rd most frequent problem outside the activities of core software product management. None of the analysed resources identified guidelines or solutions for software product managers when they experience this problem in the organisation.

As other problems previously identified may be also experienced by other teams and roles, results from this research can be widely explored within the organisation.

B. THREATS TO VALIDITY

We have conducted a methodical analysis of threats to the validity of our research using the approach proposed by Wohlin et al [26]. Threats are presented by type, in the following order: conclusion validity, internal validity, construct validity, and finally external validity.

Threats to the conclusion validity determine the ability to draw correct conclusions based on study results [26]. In this regard, a potential threat may be related to the role of the focus group moderator. As the moderator role was held by an experienced product manager, the phenomenon of ‘fishing’ may have occurred, i.e., the moderator may have influenced the outcome by steering the discussion toward her own observations. However, this threat was minimized by the presence of a second moderator, watching over methodological issues, among others.

Internal validity refers to the influences that can affect the independent variables [26]. In this context, the only issue is the list of problems that were identified in the previous survey of product managers. Focus group participants may have encountered other problems that were not included in the list of issues for discussion. However, we believe that this risk was mitigated because respondents signed up for the survey knowing the list of problems and feeling that they were affected by them.

Construct validity refers to the generalizability of experimental results to the concept or theory underlying the experiment [26]. The main risk in this regard is related to the fact that participants in the study discussed during the focus groups and evaluated the problems that had been identified in the previous study [16]. Thus, they might have been confronted with challenges that they had not encountered during their work. To minimise this risk, only product managers who declared in the recruitment questionnaire that they had experienced problems in a particular problem category were selected. We did not identify any other serious risks of this type. The focus group research technique is suitable for gathering experiences from expert groups, and this formed the basis for answering question RQ1 [24]. Furthermore, a questionnaire containing a series of simple and clear questions to which respondents answered using a widely known and well-established Likert scale was used to evaluate the solutions developed (RQ2 and RQ3).

External reliability refers to factors that limit the ability to generalise experimental results to industrial practice [26]. In this context, the main threat of both the focus group and the questionnaire, and concerns the representativeness of the experts who participated in the study – only professionals living and working in Poland participated in the survey. However, a significant number of them work in local branches of multinational corporations. Moreover, the maturity of the IT market in Poland is high enough that the threat of not being able to generalise results to the broader global market is low.

VI. CONCLUSION AND FUTURE WORK

This article explores the strategies that software product managers use to overcome the challenges they face in their work. The purpose of the study presented was to identify and evaluate appropriate solutions and guidelines that software product managers can use to overcome the main problems. This was achieved by conducting 15 focus groups with 47 software product managers to identify solutions and a set of surveys to evaluate them. These surveys focused mainly on the comprehensibility and effectiveness of the identified solutions. As a result, the article presents a set of solutions to the 5 main problems and the guidelines derived from them.

A. IMPLICATIONS FOR THEORY AND PRACTICE

We believe that both the guidelines and solutions for the most common software product management problems proposed in this study can be used in the daily work of product managers and other roles that face similar challenges when developing software products (i.e. analysts, software engineers, quality engineers, product designers). However, this will be further researched to validate them in real-world situations.

Results from the research can be also used by the authors of existing SPM assessment frameworks while making evaluations and adjustments. Recognised problems may also be used to create new assessment frameworks: the software product management process itself or the maturity of the organisation, 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 comprehensibility and effectiveness of the solutions. Using this research instrument, more solutions can be verified and the list of common and severe solutions may be extended.

The presented list of problems, guidelines and solutions together with their relative rankings may be particularly useful for inexperienced software product managers as they could not have heard and tried out the majority of the solutions yet in their career. Also, the problems and solutions may differ from one company to the other, so software product managers with experience only in one company may also benefit from reviewing a list of recommended solutions as a kind of checklist to verify their own situation.

B. FUTURE WORKS

The body of knowledge in this field was updated, which addresses many changes in the IT industry in recent years, including agile and software product management frameworks becoming more widely used. The resulting list of problems should be further researched, especially for other solutions related to software product managers that were identified.

Furthermore, it may be studied if there is any correlation between problems, solutions, company size, stage of the product lifecycle and other variables. The research may be also conducted with the top management to check what their

perspective is and how they solve the problems related to software product management, as it also affects their business. Finally, a ranking of solutions might be elaborated.

The results from this study will be further studied within our research project, especially how product managers and other roles within product development teams can utilise the whole Software Product Management Guide to influence software product management and software engineering practices in the organisation. The framework's suitability as an assessment tool for software product management maturity will also be validated through cross-team validation.

REFERENCES

- [1] C. Ebert, "The impacts of software product management," *J. Syst. Softw.*, vol. 80, no. 6, pp. 850–861, Jun. 2007.
- [2] H.-B. Kittlaus and S. A. Fricker, *Software Product Management*. Cham, Switzerland: Springer, 2017.
- [3] M. Eriksson. (2015). *The History and Evolution of Product Management*. Accessed: Jan. 17, 2022. [Online]. Available: <https://www.mindtheproduct.com/history-evolution-product-management/>
- [4] S. Haines, *Product Manager's Desk Reference*. New York, NY, USA: McGraw-Hill, 2014.
- [5] L. Gorchels, *The Product Manager's Handbook 4/E*. McGraw-Hill, 2011.
- [6] J. Sutherland and K. Schwaber. (2020). *The 2020 Scrum Guide*. Accessed: Jan. 17, 2022. [Online]. Available: <https://scrumguides.org/scrum-guide.html>
- [7] G. Steinhardt, *The Product Manager's Toolkit*. Cham, Switzerland: Springer, 2017.
- [8] M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen, "Large-scale agile transformation at Ericsson: A case study," *Empirical Softw. Eng.*, vol. 23, pp. 1–15, Oct. 2018.
- [9] H.-B. Kittlaus, "Software product management and agile software development: Conflicts and solutions," in *Software for People*. Cham, Switzerland: Springer, 2012, pp. 83–96.
- [10] A. Maglyas, U. Nikula, and K. Smolander, "What are the roles of software product managers? An empirical investigation," *J. Syst. Softw.*, vol. 86, no. 12, pp. 3071–3090, Dec. 2013.
- [11] A. Maglyas, U. Nikula, K. Smolander, and S. A. Fricker, "Core software product management activities," *J. Adv. Manage. Res.*, vol. 14, no. 1, pp. 23–45, Feb. 2017.
- [12] C. Ebert, "Software product management," *IEEE Softw.*, vol. 31, no. 3, pp. 21–24, May 2014.
- [13] G. Geracie and S. D. Eppinger, *The Guide to the Product Management and Marketing Body of Knowledge*. Product Management Educational Institute, 2013. [Online]. Available: <https://books.google.es/books?id=CJ4HngEACAAJ> and https://books.google.es/books/about/The_Guide_to_the_Product_Management_and.html?id=CJ4HngEACAAJ&redir_esc=y
- [14] C. Ebert and S. Brinkkemper, "Software product management—An industry evaluation," *J. Syst. Softw.*, vol. 95, pp. 10–18, Sep. 2014.
- [15] M. Cagan, *Inspired: How to Create Tech Products Customers Love*. Hoboken, NJ, USA: Wiley, 2017.
- [16] O. Springer and J. Miler, "A comprehensive overview of software product management challenges," *Empirical Softw. Eng.*, vol. 27, no. 5, pp. 1–38, Sep. 2022.
- [17] N. Iyer. (2019). *Surprising Stats on the Demand for Product Managers Roles in the us*. Accessed: Jan. 17, 2022. [Online]. Available: <https://medium.com/agileinsider/incredible-growth-in-demand-for-product-managers-in-the-us-but-not-necessarily-in-the-places-you-d-936fec5c1932>
- [18] (2021). *The Rise of the Product Manager in 2021*. Accessed: Jan. 17, 2022. [Online]. Available: <https://www.xcde.com/blog/2021/05/the-rise-of-the-product-manager-in-2021>
- [19] S. Hyrynsalmi, A. Suominen, and M. Seppänen, "A bibliographical study of software product management research," in *Proc. IEEE Int. Conf. Eng., Technol. Innov. (ICE/ITMC)*, Jun. 2021, pp. 1–8.
- [20] A. Maglyas, U. Nikula, and K. Smolander, "Lean solutions to software product management problems," *IEEE Softw.*, vol. 29, no. 5, pp. 40–46, Sep. 2012.
- [21] C. Ebert, "Managing software products in a global context," in *Proc. IEEE/ACM 13th Int. Conf. Global Softw. Eng. (ICGSE)*, May 2018, pp. 64–71.
- [22] T. Wagenblatt, "Software product management fundamentals," in *Software Product Management*. Cham, Switzerland: Springer, 2019, pp. 3–58.
- [23] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting empirical methods for software engineering research," in *Guide to Advanced Empirical Software Engineering*. Cham, Switzerland: Springer, 2008, pp. 285–311.
- [24] J. Kontio, J. Bragge, and L. Lehtola, "The focus group method as an empirical tool in software engineering," in *Guide to Advanced Empirical Software Engineering*. Cham, Switzerland: Springer, 2008, pp. 93–116.
- [25] J. Kontio, L. Lehtola, and J. Bragge, "Using the focus group method in software engineering: Obtaining practitioner and user experiences," in *Proc. Int. Symp. Empirical Softw. Eng.*, Apr. 2004, pp. 271–280.
- [26] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Cham, Switzerland: Springer, 2012.



OLGA SPRINGER received the master's degree in computer science. She is currently pursuing the Ph.D. degree with the Department of Software Engineering, Faculty of Electronics and Telecommunications, Gdańsk University of Technology. She was a Senior Product Manager of Atlasian and leads the biggest Polish community on software product management named ProductVision.pl. Her research interest includes software product management.



JAKUB MILER received the Ph.D. degree. He is currently an Assistant Professor with the Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Poland. He was the Quality Manager of Argevid Ltd., a spin-off that offers assurance and conformance analysis software. He supervises the IT management student research group and co-organizes the annual beIT Software Engineering Conference. His research interests include software project and product management, risk management and human factors in particular, agile software development, trust to information systems, and the social aspects of information technology.



MICHAŁ WRÓBEL received the Ph.D. degree in computer science from the Gdańsk University of Technology, Gdańsk, Poland, in 2011. He is currently an Assistant Professor with the Gdańsk University of Technology. Since 2006, he has been with the Department of Software Engineering, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology. His research interest includes modern approach to software development management, with a particular focus on the role of human factors in software engineering. He is also a member of the Emotions in HCI Research Group, where he conducts research on software usability, affective computing, and software management methods.

...