

**This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.**

**Author(s):** Melegati, Jorge; Wiese, Igor; Guerra, Eduardo; Chanin, Rafael; Aldaej, Abdullah; Mikkonen, Tommi; Prikladnicki, Rafael; Wang, Xiaofeng

**Title:** Product managers in software startups : A grounded theory

**Year:** 2024

**Version:** Published version

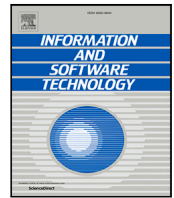
**Copyright:** © 2024 The Author(s). Published by Elsevier B.V.

**Rights:** CC BY 4.0

**Rights url:** <https://creativecommons.org/licenses/by/4.0/>

**Please cite the original version:**

Melegati, J., Wiese, I., Guerra, E., Chanin, R., Aldaej, A., Mikkonen, T., Prikladnicki, R., & Wang, X. (2024). Product managers in software startups : A grounded theory. *Information and Software Technology*, 174, Article 107516. <https://doi.org/10.1016/j.infsof.2024.107516>



## Product managers in software startups: A grounded theory

Jorge Melegati<sup>a,\*</sup>, Igor Wiese<sup>b</sup>, Eduardo Guerra<sup>a</sup>, Rafael Chanin<sup>c</sup>, Abdullah Aldaej<sup>d</sup>,  
Tommi Mikkonen<sup>e</sup>, Rafael Prikladnicki<sup>c</sup>, Xiaofeng Wang<sup>a</sup>

<sup>a</sup> Free University of Bozen-Bolzano, Bolzano, Italy

<sup>b</sup> Technological University of Parana (UTFPR), Campo Mourão, Brazil

<sup>c</sup> PUCRS, Porto Alegre, Brazil

<sup>d</sup> Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia

<sup>e</sup> University of Jyväskylä, Jyväskylä, Finland

### ARTICLE INFO

#### Keywords:

Product manager  
Software startup  
Product management  
Agile software development  
Socio-technical systems  
Grounded theory

### ABSTRACT

**Context:** Defining and designing a software product is not merely a technical endeavor, but also a socio-technical journey. As such, its success is associated with human-related aspects, such as the value users perceive. To handle this issue, the product manager role has become more evident in software-intensive companies. A unique, challenging context for these professionals is constituted by software startups, emerging companies developing novel solutions looking for sustainable and scalable business models.

**Objective:** This study aims to describe the role of product managers in the context of software startups.

**Method:** We performed a Socio-Technical Grounded Theory study using data from blog posts and interviews.

**Results:** The results describe the product manager as a multidisciplinary, general role, not only guiding the product by developing its vision but also as a connector that emerges in a growing company, enabling communication of software development with other areas, mainly business and user experience. The professional performing this role has a background in one of these areas but a broad knowledge and understanding of key concepts of the other areas is needed. We also describe how differences of this role to other lead roles are perceived in practice.

**Conclusions:** Our findings represent several implications for research, such as better understanding of the role transformation in growing software startups, practice, e.g., identifying the points to which a professional migrating to this role should pay attention, and the education of future software developers, by suggesting the inclusion of related topics in the education and training of future software engineers.

### 1. Introduction

Software Engineering (SE) concerns not only *how* to develop systems, such as coding and architecture, but also *what* these systems will perform. An example is the long-standing field of requirements engineering (RE). As Cheng and Atlee [1] argue, the problems in this field are distinct from others in SE “because requirements reside mainly in the problem space whereas other software artifacts reside primarily in the solution space”. Recently, a role focusing on this aspect has become more evident — the product manager (PM). These actors perform several tasks, collecting information from users through several means, such as in-person interactions and data usage collection, “to figure out what to build, define requirements, and write functional specifications” [2]. In other words, to achieve technical goals as elicitation, specification, and validation of requirements, product managers must collect information from developers, users, and other stakeholders,

empathizing with these actors and handling conflicts among them to lead to the product’s success. Hence, product management is essentially a socio-technical phenomenon defined by Hoda [3] as when “social and technical aspects are interwoven in a way that studying one without due consideration of the other makes for an incomplete investigation and understanding”.

A context particularly challenging for product management is constituted by software startups. Given that different definitions have been used in the software engineering literature [4,5], it is essential to define the term clearly. In this paper, we define software startups as companies that develop new, innovative, software-intensive products or services under constrained resources and time, searching for sustainable and scalable business models [6]. This unique context is associated with a high level of uncertainty, especially regarding what customers and users want and are willing to pay for [4,5]. Besides unstable requirements and constant strategic changes [7], PMs in software startups

\* Corresponding author.

E-mail address: [jorge.melegati@unibz.it](mailto:jorge.melegati@unibz.it) (J. Melegati).

<https://doi.org/10.1016/j.infsof.2024.107516>

Received 6 July 2023; Received in revised form 26 April 2024; Accepted 10 June 2024

Available online 13 June 2024

0950-5849/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

must face other issues, such as the lack of users to collect feedback from [8], the scarcity of resources [9], and an accelerated development process [9,10]. In this context, conventional practices might not suffice and practitioners and academics have proposed novel methodologies, based on experimental or hypotheses-driven approaches [11–13]. These proposals suggest that techniques related to product management in software startups are distinct from consolidated companies. Other aspects also indicate this direction such as small teams and financial constraints. Although several studies focused on RE in the context of software startups, e.g., [14–16], and others on the PM role in software projects, e.g., [17–19], to the best of our knowledge, no studies concentrated on the PM role in software startups. In this paper, we aim to fill this gap by describing several aspects associated with the PM role in software startups. The following research question represents this goal:

**RQ. What is the role of the product manager in software startups?**

To answer the research question, we followed the Socio-Technical Grounded Theory approach [3] using the structured mode in the advanced stage to develop the theory. As usual in Grounded Theory studies, data collection and analysis were iterative and interleaved. Data consisted of blog posts written by practitioners describing their activities as PMs in software startups and interviews with members of startup teams performing the PM role. In the process of selecting blog posts to be included in the study, we employed an automatic summarization based on PositionRank [20], an unsupervised text summarization algorithm, to create abstracts for these documents and, consequently, reduce the amount of work in the selection process. We described this approach in detail in a previous paper [21]. The interviews started at the end of the basic stage and continued in the advanced stage to allow theoretical sampling. The result consists of a theory of the product manager in software startups describing the role as multidisciplinary and ill-defined, dependent on the context and stage of the company. Although heterogeneous, the responsibilities of this role concentrate not only on software development but also on other areas, especially business and design. Besides that, the role is a connector element, facilitating communication among these different areas and guiding them using a vision for the product. In this regard, this agglutinating is more evident as startups grow and areas start to specialize rather than in the early stages when few people perform many tasks.

The contributions of this study are four-fold. First, we understand the PM role in software startups better by describing it as a multidisciplinary and ill-defined role with a connecting function in a growing company in which people start to silo functional teams. Then, we describe the needed skills for these professionals, representing guidelines for the education and training of software engineering students to fulfill these positions. Third, we present the use of blog posts to support a Grounded Theory study, in contrast to similar studies in software engineering research, which are heavily based on interviews. Finally, we extend the evaluation of the automatic summarization approach for blog posts by employing the selected documents in the study.

The rest of this paper is organized as follows. Section 2 presents the basic concepts and a lean literature review. Section 3 describes how we performed the study. Section 4 presents our results, namely a theory on the PM role in software startups. In Section 5, we discuss how our results relate to the existing literature and the implications to research and practice. Finally, Section 7 concludes the paper.

## 2. Background and related work

In this section, we present the key concepts related to our study. These include product manager, introduced in Section 2.1, and software startups, presented in Section 2.2.

### 2.1. Product manager

In the seminal marketing book, Kotler [22] defines a product “as anything that is offered to a market for attention, acquisition, use, or consumption that might satisfy a want or need”. The term encompasses not only physical objects but also “services, persons, places, organizations, ideas or mixes of these entities”. The author also defines services as “products that consist of activities, benefits or satisfactions offered for sale that are essentially intangible and do not result in the ownership of anything”. Based on this definition, Kittlaus and Clough [18] define a product as a “combination of (material and/or intangible) goods and services, which one party (called vendor) combines in support of their commercial interests, to transfer defined rights to a second party (called customer)”. Then, a software product is a “product whose primary component is software” [18,23].

The term “product manager” is credited to Neil McElroy, who proposed the idea of a role responsible for a product to the executive team at Procter & Gamble in 1931 [2]. The role has been incorporated into software engineering to fill developers’ difficulty to “meet technical demands and create a cohesive product for users” [2]. Chisa [2] also mentioned that a PM is a person who talks to users or customers to figure out what to build by writing functional specifications. Moreover, PMs need to be involved in the whole product life cycle: before the first line of code is written, during the software development process, and after the product goes into production. In the XP (Extreme Programming) proposal, Beck [24] describes what is expected from a PM in a team employing the method. They should “write stories, pick themes and stories in the quarterly cycle, pick stories in the weekly cycle, and answer questions as implementation uncovers under-specified areas of stories”.

Regarding the context of software-intensive companies, it is essential to mention the International Software Product Management Association<sup>1</sup> (ISPMA<sup>2</sup>), a non-profit organization whose goal is “to foster software product management across industries” [25]. Specifically, the organization recommends a handbook [25] as a knowledge reference for the role of software product manager. Nevertheless, the existence of recommended practices does not guarantee that they are followed in practice. Actually, the differences between the state-of-the-art and the practice of SE have been a concern since early papers [26]. Researchers have investigated the state-of-the-practice for several areas of SE, such as requirements engineering [27] and software architecture [28]. This issue is even more prominent in software startups, in which researchers have observed the lack of use of prescribed practices [8,16,29]. Besides that, the existence of such guidelines, first, do not guarantee that they are the only effective and efficient practices, and, second, they were not tailored for the context of software startups which, given its particularities, might require specific practices.

Studies describing how the PM role is in practice in the context of software-intensive companies portray a varied picture. Based on 17 interviews and supporting documentation from 13 companies, Maglyas et al. [30] developed a theory with four categories regarding the role of PMs: influence on the product, authority, access to resources, and impact on collaboration. Based on them, the authors identified four stereotypical roles: experts, strategists, leaders, and problem solvers. These results are interesting. However, the study relied on established and large companies: the youngest company had four years at the time of the paper’s publication, 12 out of the 13 companies had more than 101 employees, and the remaining had between 11–50. Springer and Miler [31] performed another study comparing PMs across four levels of company size: micro, small, medium, and large businesses. They described competencies and objectives depending on the size and concluded that the organization’s size influences the role. Among the

<sup>1</sup> <https://ispma.org/>.

<sup>2</sup> “ISPMA” is a registered trademark of ISPMA e.V.

ten companies investigated, three were classified as “micro/small” or “micro”, i.e., less than 10 employees, and the other two as “small”, i.e., with between 11 and 50 employees. However, there are no details about the innovativeness of the products developed by these companies, which we could consider as startups according to our definition, nor a clear consideration of this aspect of the results. Kittlaus and Fricker [32, p. 37] describe four scenarios for software products regarding life cycle phases, i.e., new product development or existing product evolution, and runtime environment, i.e., vendor or customer-controlled. The scenario characterized by new product development in a vendor-controlled scenario, called Powerboat by the authors, is similar to startups. The authors recommend that, in this scenario, PMs should work closely with potential customers and marketing, and they should support short-term experimentation. Despite the value of this prescriptive description, there is no study focused on startups to understand if these pieces of advice are being followed in practice. Other studies focused on the challenges related to software product management [33] or its conflicts with agile [34].

Besides these studies, the role of PMs in software companies is often mentioned when investigating related aspects. In a study on product backlogs at Pivotal, a large software development company, Sedano et al. [35] describes product management as “the voice of the business” and mentions that its main role is “to advocate for the business’s needs and facilitate prioritization of the team’s backlog”. The company employs a software development process consisting of two tracks: one led by product designers and the other by developers. In this context, the PM is a boundary spanning the two tracks, bridging the gap between the two tracks [35]. Specifically in software startups, in a study on requirements engineering, Melegati et al. [16] observed that PMs are responsible for most requirements engineering activities, especially elicitation, and prioritization, acting as a proxy for the potential users and customers. They also observed that a team performing this role is common in software startups [16]. Still, in a study on requirements engineering in software startups, Gralha et al. [14] concluded that PMs are responsible for managing the product backlog.

The PM role is related to the product owner in Scrum. In the method proposal, Sutherland and Schwaber [36] describe the product owner as the “responsible for maximizing the value of the product resulting from work of the Development Team”. The authors acknowledge that the way this is done may vary widely across organizations, teams, and individuals. Given the prevalence of agile methods in contemporary software engineering, the product owner has been the topic of several studies, such as comparing theory and practice [37], how they are tailored to large-scale projects [38], and their importance to communication [39]. Specifically, Kittlaus [34] discussed conflicts between software product management and agile software development, giving a focus on the product owner role. The authors identified several tasks associated with PMs that are not related to the product owner role and, in general, product managers are closer to the businesses and the customers, while the product owner is closer to the development and technology. However, the author acknowledges that in small organizations, the PM often assumes the product owner role.

In summary, the PM role encompasses different responsibilities, and what is executed highly depends on the company. Hence, it is valuable to investigate this role in software startups, given how crucial these activities are to these companies.

## 2.2. Software startups

Software startups have generated increasing interest in software engineering research [5]. These companies develop innovative, software-intensive products to which they search for a sustainable and scalable business model [6]. As emerging organizations, software startups are expected to experience several changes in their internal processes as the team and the number of customers grow. Hence, since early papers on the topics, e.g., [40], researchers have proposed maturity models

to describe the evolution of software startups. The state-of-the-art life-cycle model for software startups was proposed by Klotins et al. [10], based on the original model of Crowne [40], and is composed of four stages: inception, stabilization, growth, and maturity. The inception stage starts with the initial idea and ends with the first release of the product to customers. Then, startups enter the stabilization stage, preparing the product to scale regarding technical and operational aspects. After this preparation is ready, in the growth stage, the startup goal is to gain the desired market share focusing on marketing and sales. Finally, the organization transitions into an established company during the maturity stage. These stages can be summarized into two blocks: (1) inception and stabilization (or the early stages), (2) growth and maturity (or the later stages). In the early stages, the startup aims to find a relevant problem and a feasible solution. In the later stages, the focus is on marketing and improving the company’s efficiency [10]. However, this process might not be linear, and when facing specific challenges, startups may change some critical aspects of the product and, consequently, return to a previous stage.

Software development in startups is characterized by a sped-up process under a severe lack of resources, leading to low priority for product quality and accumulated technical debt [9]. The importance of defining the product in such a complex context means that requirements engineering (RE) is the central activity in software startups [29]. Based on an analysis of 88 startups, Klotins et al. [29] concluded that “most engineering challenges in startups stem from inadequacies in requirements engineering”. This fact has inspired several studies focusing on how these companies handle requirements engineering activities. Gralha et al. [14] investigated the evolution of RE practices in software startups. They proposed a grounded theory where RE practices are described through six dimensions: requirements artifacts, knowledge management, requirements-related roles, planning, technical debt, and product quality. These dimensions evolve based on some turning points, such as the number of clients or employees. For instance, regarding requirements artifacts, at the beginning of its life, a startup is implementation-oriented. Once the number of clients, remote workers, and employees rises, the team moves to a more user-oriented approach. Finally, when the number of features and input from the customers increases, the team reaches richer, traceable descriptions.

Tripathi et al. [15] performed a mixed-method study consisting of multi-vocal literature and a case survey to investigate five aspects of RE sources: requirements, elicitation methods, requirements documentation, prioritization and management, and validation. The results consisted of a frequency analysis for different techniques. Melegati et al. [16] also investigated RE in software startups through interviews with Brazilian practitioners. They concluded that software startups do not follow a unique set of RE practices and they change during the startup’s lifetime. In their model, practices are determined by a set of factors: founders, software development manager, developers, market, business model, and startup ecosystem. The authors also concluded that startups adopt practices similar to agile teams with the difference of not having an accessible customer.

In the aforementioned studies, product management is not depicted in detail. Regarding the requirements-related roles, Gralha et al. [14] mention that “a product manager or a product owner is responsible for creating and prioritizing the product backlog” and that “companies with multiple products will eventually have a director of products to oversee the various product managers and determine product priorities”. Melegati et al. [16] observe that many startups they interviewed, especially those with products targeted to a large market, had a product management team. They observe that “this team was not composed of developers although its components can have a technical background, but also business or marketing profiles”. Regarding their responsibilities, they mention that PMs “are responsible for most of the RE stages, particularly elicitation and prioritization”. It is also essential to mention that some authors, such as Blank [12] and Ries [11], stress the need for specific methods for startups, which are generally focused

on experiment-driven development and, consequently, focus more on hypotheses about the product than on requirements [41]. Research has shown that practitioners are employing several techniques to handle these hypotheses, including elicitation and prioritization [13].

In summary, the authors of this study could not find any work in the literature specifically focused on the PM role and product management in the context of software startups. Even though there are some proposals, such as the one by Kittlaus and Fricker [32], of how PMs should act in the context of new product development, there is no study investigating how the roles are performed in practice.

### 3. Research method

In their seminal paper, Bostrom and Heinen [42] describe a socio-technical system as a work system “made up of two *jointly* independent, but correlative *interacting* systems - the social and the technical” (emphasis in the original). While the technical perspective concerns “processes, tasks, and technology”, the social one regards “the attributes of people (e.g., attitudes, skills, values), the relationships among people, reward systems, and authority structures”. As described above, the product manager is a role involving both technical aspects, as processes and technology, but also social ones, such as skills, relationships, and authority structures. Therefore, to achieve our goal of investigating the role of PMs in software startups, we performed a Grounded Theory (GT) study following Hoda’s Socio-Technical GT (STGT) guidelines [3]. This choice is motivated by the fact that the method is tailored for software engineering research, especially considering its socio-technical aspects and its description contains all the operationalization details required to properly employ the method. Besides that, we observed the core features identified by Stol and Fitzgerald [43] and stressed by the ACM SIGSOFT empirical guidelines for GT [44]: limit exposure to literature, treat everything as data, immediate and continuous data analysis, theoretical sampling, theoretical sensitivity, coding, memoing, constant comparison, memo sorting, cohesive theory, and theoretical saturation. The method was chosen given the socio-technical nature of the phenomenon and the fact that STGT was developed for software engineering research.

The Socio-Technical Grounded Theory method comprises two key stages: a basic and an advanced one. The basic stage consists of a lean literature review, study preparation and piloting, and iterations of basic data collection and analysis. In this step, concepts, categories, and relationships emerge. The advanced stage consists of theory development and could be implemented with one of two modes: emergent and structured. The emergent mode is inspired by Glaser’s GT version while the structured version is represented by the mode of Corbin and Strauss [45].

We followed the guidelines by Hoda [3] for a lean literature review. According to the author, the step is “a lightweight and high-level review performed early in the study, during the basic stage, to identify research gaps and motivate the need for a study”. It is important to stress though that, in this stage, we avoided inspecting in detail the papers presenting the studies that could bias our analysis, following the general guidelines for GT studies to limit the exposure to literature, since our goal is to develop a new theory to explain the phenomenon rather than validating an existent one. Section 2 above presents the papers we analyzed in this stage. Besides presenting key concepts for our study, such as software startups, we observed the lack of studies focused on the role of product managers in software startups and motivated why this particular context presents specific challenges for the role.

In this study, data were collected from two sources: (1) blog posts authored by PMs and (2) interviews with practitioners. Section 3.1 describes the data collection in detail. In the basic stage, most of the data consisted of blog posts. Still, following the suggestion of an interleaved data collection and analysis, after each document was included in the analysis, we performed a constant comparison between

the recent codes emergent from open coding to those already present in the analysis. In this study, we considered blog posts as data [46–48], i.e., as descriptions of what happens in software startups, rather than literature. After we ended up performing the first analysis of the blog posts, we started performing interviews with practitioners. After a couple of interviews, it was evident that: (1) the PM is on the boundary of several areas: software development, business, and design/user experience; (2) the importance of communication; and (3) the evolution of the role depending on the startup stage.

Based on these central concepts, we followed a structured mode of theory development. Hoda suggests that the researchers can explore “pre-defined theoretical templates”, such as coding paradigm, conditional matrix, or Glaserian six C’s [3]. Researchers could also “use their own theoretical structure emerging from the basic stage”. Therefore, we came up with a structure for the emergent theory. This structure was used “to guide further data collection, analysis, and theory development” [3]. Specifically, it pointed us to the need to interview for other roles in early-stage software startups. The analysis helped us to better group the categories along axes. Fig. 1 illustrates our research design.

#### 3.1. Data collection

##### 3.1.1. Blog posts

To obtain data from a large pool of startups, in the first iteration of data collection we gathered blog posts. In these statements, practitioners describe factual stories, analogies, and examples to support their beliefs [46]. Recently, blog posts have been employed in SE studies [47], e.g., [13,49–51]. Among the benefits of this data source are insights into software practice, research clarity and transparency, and help bridge research and practice [47]. Blog posts are sometimes called a form of gray literature [52]. This type of literature encompasses published documents that did not go through formal quality control mechanisms as in the white literature [53]. However, we agree with Kitchenham et al.’s recent argument [48] that blog posts are not gray literature but could be used as data for primary studies. In any case, to guide this step of data collection, we followed Garousi et al.’s [52] guidelines for gray or multi-vocal literature reviews. Although we are not considering blog posts as literature but rather as data, these guidelines could help us to gather a wide variety of experiences.

We searched sources using a general web search engine, i.e., Google, in June 2021. To obtain varied results, we employed a query string only filtering startups, i.e., not considering the term “software” to avoid excluding posts that potentially have useful data. Regarding product managers, we also considered the term “product management”. Hence, the final query string was:

(“product management” OR “product manager”)  
AND (startup OR start-up)

To avoid the interference of cookies and previous search history, we performed the search in a Google Chrome incognito window. The result consisted of 434 results that we saved in a CSV file using SEOquake.<sup>3</sup> Given our interest in practitioners’ views, we inspected the links and removed those that were not blog posts. Examples of excluded items are job position ads, podcasts, videos, and book descriptions. We saved locally the main text from the included links. After this step, our dataset consisted of 127 documents.

The next step was to select posts that could help us to answer the research question, i.e., that contain real experiences of PMs in software startups. Since blog posts do not have abstracts as scientific articles, this task would consist of reading the whole text. To reduce the overhead over the researchers, we decided to employ PositionRank,

<sup>3</sup> <https://www.seoquake.com/index.html>.



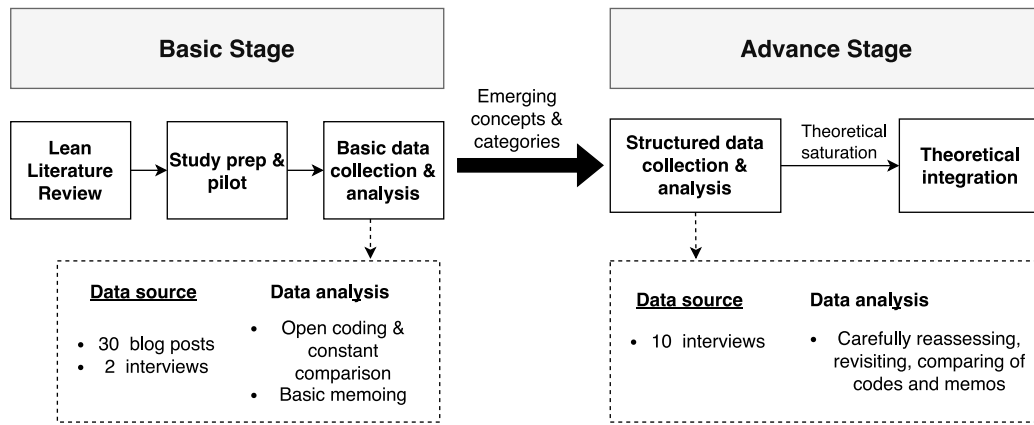


Fig. 1. A summary of the research design.

a summarization algorithm, to create abstracts for the blog posts. PositionRank is a modification to consider the position of a word in the document to the PageRank algorithm when applied to the text where a graph representing the text is used to identify the key phrases by graph centrality measures. Then, we divided the set of documents into two, and each researcher read the full text for the documents in one set and the generated summaries from the other set. In such a way, each document was analyzed in full by one researcher and as a summary by another.

Then, we defined four inclusion criteria regarding three aspects: type of evidence, context, and subject. Regarding the type of evidence, Rainer [46] differentiates the information from gray literature sources in primary, i.e., the author's personal experience clearly informs that, and secondary, i.e., that is not based on the personal experience and is probably formed by indirect sources such as peers. According to the author, primary information is "most closely connected to evidence" and, thus, is more useful for argumentation purposes. Given our interest in using blog posts as primary data describing the practice of PMs, we chose to select only primary information. Thus, the first criterion regarded if the document described a real experience. We excluded, for example, posts simply presenting the author's opinion without presenting any account of real events in which the author participated. For example, in an article,<sup>4</sup> the author simply lists several tools that could be used by a PM in a startup but he does not describe a situation in which himself used any of them. Regarding the context, our focus was software startups. To operationalize the analysis of this aspect, we divided it into two criteria. First, the statement should refer to the context of startups in general what was then considered our second criterion, and then the document should focus on a software development context, that is our third criterion. Finally, regarding the subject, the focus was on the description of the PM role, therefore this aspect was the fourth criterion.

After this process, the researchers agreed that 19 blog posts should be included in the analysis and 73 not included. Out of the 73 documents excluded at this step, 50 were considered not describing a real experience, 16 were not related to a startup context, 22 were not describing a software-related context, and finally, 11 were not describing the PM role. The total sum is larger than 73 because some blog posts have failed in multiple criteria. On the remaining 35 documents, they were either undecided or had conflicting answers. Then, for each one of these documents, the researcher who had read the summary read the full text, and they discussed reaching an agreement. Out of the 35 documents discussed, 11 were accepted. Out of the 24 rejected, 7 were considered not describing a real experience, 6 were not related

to startups' context, 16 were not describing a software-related context, and 10 were not related to the PM role. Once again, a document could have been rejected for multiple reasons. After this process, a total of 30 documents were included in this study, which are listed in Appendix A. Fig. 2 presents a summary of this process. Since there might be license problems in making blog posts available in replication packages [48], we only provide the URLs of the blog posts.

### 3.1.2. Interviews

To select practitioners to be interviewed, we followed a purposive sampling, a form of non-probability sampling whose goal is "to find accessible, information-rich cases, sites, organizations or contexts from which researchers can learn about their topic of study" [54]. By taking advantage of the geographical distribution of the authors, we identified participants from five countries in North and South America, Europe, and the Middle East. The interviewees represent startups operating in different markets and at different stages of maturity. In summary, as shown in Table 1, we aimed to consider startups as diverse as possible, the so-called "heterogeneity sampling" [54]. We followed a semi-structured approach for the interviews using a pre-defined interview guide [55], presented in Appendix B. It was divided into four sections: individual, company, role, and conclusion. In the first section, we asked about the background of the interviewee including if she had a formal degree and in which area, and her previous working experience either as a PM or in other roles. Then, we asked for the details about the company, including how long it existed, its size, and its stage of development at the time of the interview. We also asked for a description of the product under development and who are the potential customers and users. In the third section, we asked the interviewee to describe her duties and responsibilities, including those that she considers are the most important, the tools she uses, if she has a daily routine, and if she had experience as a PM in another company, if there are differences and what they are. Finally, we asked if the interviewee had anything to add or any comments and feedback. The interviews were conducted online through video calling and were recorded and transcribed. We performed ten interviews when theoretical saturation was reached. Theoretical saturation is a point when "further collection does not generate new or significantly add to existing concepts, categories, or insights" [3]. After each interview, the data obtained was incorporated and analyzed, compared with previously collected data. Even though new codes emerged in the analysis of the last interviews, these codes did not alter the theory structure nor their categories, just being more examples of some categories. This aspect became clearer after the sixth interview, even though none of the final categories of the theory is supported solely by interview data but also by blog posts' data. However, as until that, there was a geographical concentration on the interviews, we decided to perform more to increase the variability. The interviews were conducted during the first semester of 2022, and their duration ranged from 28 min to one hour, with an average of 43 min. Table 1 presents details of the interviewees.

<sup>4</sup> <https://theproductmanager.com/tools/tools-for-startup-product-managers-build-measure-learn/>.

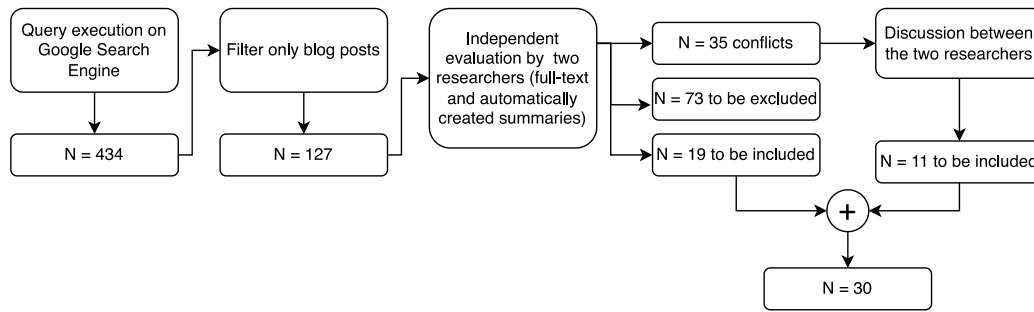


Fig. 2. A summary of the selection of blog posts.

Table 1

List of the interviews performed.

	Interviewee position	Company stage [10]	Market	Country	Company life time <sup>a</sup> (years)	Time in the company <sup>a</sup> (years)	Other information
11	Product manager	Growth	Communication	Brazil	6	1	Previous experience as founder in another startup.
12	Product manager	Mature	Logistics	USA	8	1.5	–
13	CEO	Stabilization	Health	Brazil	2	Founder	No profit company.
14	Product manager	Stabilization (internal startup)	SaaS for e-commerce	Brazil	1	1	–
15	Product manager	Mature	Tourism	Brazil	11	1.5	Previous experience as founder in another startup.
16	CEO	Stabilization	SaaS for sales	Brazil	3	Founder	–
17	Head of engineering	Growth	Finance	Germany	5	2.5	The company went through a major pivot 3 months the interviewee joined the company
18	Project manager	Stabilization	SaaS for health	Saudi Arabia	4	From the beginning	–
19	Product manager	Stabilization	Services	Saudi Arabia	2	Founder	–
110	Founder	Stabilization	Software	Finland	3	Co-founder	–

<sup>a</sup> Approximate.

### 3.2. Data analysis

The Socio-Technical Grounded Theory method is divided into basic and advanced stages [3]. In the basic stage, we applied open coding, constant comparison, and memoing. After emerging some concepts (in the advanced stage), we followed a more structured data analysis to reassess, revisit, and compare codes and memos carefully. We used NVivo12<sup>5</sup> to support our coding process. All data were imported to the tool, which helped us to keep track of codes and their corresponding text. It also supports organizing or grouping codes into categories and sub-categories. At the end of the analysis, six main categories emerged, divided into 18 subcategories, that are available in the supplemental material, including, for each one of the main codes, an example excerpt from data and in which sources it was identified.

**Member checking.** After we reached a final theory, we performed a member-checking procedure. We prepared an email with a summary of our results and sent it to all the interviewees who participated in the study. We gave them one week to answer, and we sent a reminder to those who had not answered after five days. Out of the ten interviewees, we received three replies containing short comments claiming that they agreed with the results.

**Supplemental package.** To improve the replicability of our results, we provide a supplemental package<sup>6</sup> containing all the links

returned by the Google Search Engine and how the researchers rated them, leading to inclusion and exclusion. However there might be license problems with making blog posts available in replication packages [48], therefore we only provide the URLs of the blog posts. It was impossible to make the interviews' transcriptions available since there was sensible commercial information about the startups. We also provided the categories, categories, and codes identified in the analysis with examples of coded excerpts.

### 4. A grounded theory of product managers in software startups

As the result of our analysis, we reached a grounded theory describing the PM role in software startups (the main categories are in bold):

A **product manager** in software startups is a generalist, multidisciplinary, and, quite often, ill-defined role. This person, or sometimes a team that shares the responsibilities, is accountable for the product, often referred to as a mini-CEO. We grouped this role's **responsibilities** in areas that include functional areas, i.e., software product development, business, and user experience/design, but also three other coordinating functions. First, they are responsible for communication, acting as a *bridge*, connecting the different areas of the startup but also the team with customers and users. Second, PMs should perform a function of leadership, guiding the connected team to where the team should go, i.e., achieve the product vision. Finally, the PM has also management duties, coordinating and facilitating teamwork. Given the generalist, multidisciplinary nature of

<sup>5</sup> <https://www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home>.

<sup>6</sup> <https://doi.org/10.5281/zenodo.10070156>.

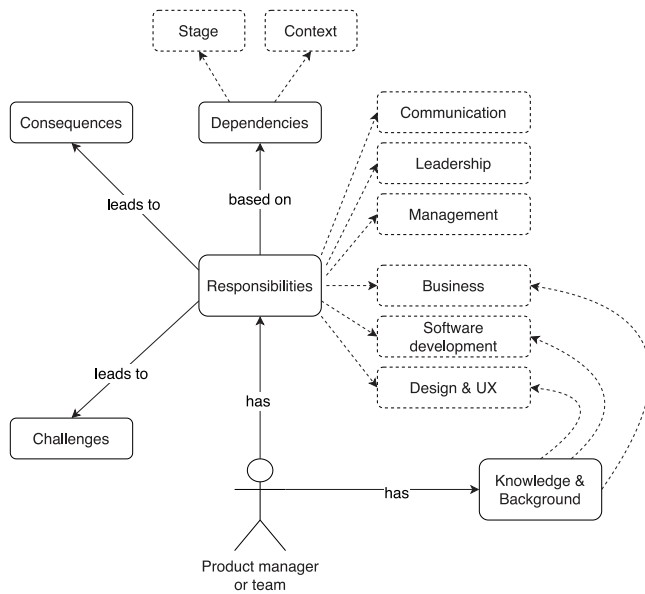


Fig. 3. A grounded theory on product management in software startups. Solid-line boxes and arrows represent core categories and their relationships. Dashed-line boxes represent subcategories linked to the core categories with dashed arrows.

the role, a person fulfilling the role of PM generally comes from one of the connected fields, i.e., she has a **background** in either software development, business, or design. However, the PM must have a broad **knowledge**, including some familiarity with the other areas. Performed tasks vary based on some **dependencies**: the stage of the product and company and the context, including, for instance, the product and the team. Finally, the existence of the PM is associated with some **consequences** and **challenges**.

Fig. 3 depicts the main categories of the theory, which are described in detail in the following sections. Below, we use bold to represent subcategories and italics to present some of the most common codes. We use excerpts of data, in italics and in quotes, either from the blog posts or from the interviews, to illustrate the results.

#### 4.1. Product manager

The product manager **role** is a *generalist, multidisciplinary role* encompassing several areas, including not only software engineering but also interface design, user experience, and business aspects of the product. This description is present in many of the analyzed blog posts. For instance, B3 mentions “the multi-disciplinary nature of the role encompassing design, engineering, marketing, and sales”. Related to that, it is considered an *ill-defined role*, i.e., without a clear definition of what are the responsibilities associated with the role. In the words of the author of B18, “not everyone has the same idea of what a product manager does”. Consequently, the PM is seen as someone who does not create direct value as I4 mentioned: “[I always say that] my job is [...] effectively to do nothing”. However, a clear aspect of the role is the *accountability for the product*, i.e., the PM decides several, if not all, aspects of the product and is responsible for its results. As B7 mentions: “product managers are often called the mini-CEO, as they are, indeed responsible for every aspect of a product [...]”.

Besides role characteristics, the analyzed sources described several aspects of the PMs’ **way of working**. They are *data and feedback driven*, i.e., to make decisions, they usually rely on product usage data or on users’ feedback. As B29 mentions: “you have intuition, and you have your experience, but your notions about what your customers want - they’re just notions. They’re not true or false until you put them to the

test”. However, this is not always the case as *intuition* has been also mentioned, especially in early stages of a startup. To cope up with time and resources constraints, PMs in startups should work with *speed*. This issue is summarized in B5 as: “the sooner you know their opinions about your product, the earlier you can adjust your plans in terms of resources and the higher likelihood of reaching your next milestone of hitting your next goal before you run out of money”. A PM also has to *trust others* especially in the area in which she is not an expert.

We also identified several personal **qualities** associated with professionals fulfilling this role. The most mentioned were: *being adaptive and flexible*, *good communicators*, and *empathetic*. These qualities are associated with the responsibilities PMs have but also to cope with the challenges of the role and of the context in which they work.

#### 4.2. Responsibilities

Although the specific responsibilities of each PM vary, they can generally be grouped into categories of two natures: functional or organizational. Functional responsibilities create direct value by being actively involved in product development and are classified in three categories: **software development**, **business**, and **user experience and design**. Organizational responsibilities are related to coordinating functions and are classified in **communication**, **leadership**, and **management**.

Among all the identified duties, the most important is probably **communication** as mentioned by several interviewees when asked. In this regard, PMs facilitate the interaction and carry information between the different areas of the startup and also between the startup’s internal, i.e., team, and external, i.e., users, customers, and other stakeholders. This connection happens in several layers not only by bringing requirements and constraints from the different areas but also by connecting people and acting as a translator between the different vocabularies used by the groups/areas. In this regard, the PM can be seen as a *bridge* gluing together the different areas of the startup, helping them to work in a coordinated manner to reach a common goal. In the words of B4: “the PM generates bridges of communication between stakeholders, keeping the focus on what is really important”. In this regard, they *work close to founders and management*, *communicate with other teams*, mainly to *communicate the vision to the team*.

As a consequence of being a bridge across different areas, PMs perform a **leadership** role. As B23 puts: “your role is to mold a product vision and to build a team aligned around that vision so everyone will have focus and purpose”. Thus, they must *build trust and relationships* and *understand and support other teams’ work*. The PM has also **management** duties. First of all, she must *facilitate the team’s work to keep the rhythm*, that is, identifying problems, removing obstacles, and finding solutions to help the team to keep a constant pace. They are responsible to *ensure that the work is done according to the plan*. Quite often, PM *define processes*, *improve*, and *document* them. In some cases, they are even responsible to coordinate the development team, including not only software engineering but also designers.

Besides connecting the different areas of software development, business, and design/UX, the PM generally performs tasks associated with each one of these areas. Regarding **software development**, these tasks are, for instance, *deciding the features to build*, *creating specifications*, and *prioritizing*. This person is responsible for *gathering ideas and requirements* and *validating* if they are worth implementing. PMs with a technical background might *influence technical decisions* too. Regarding **user experience**, the PM should *gather the feedback* and use it to *understand users and customers*. As B1 mentions: “product management is about letting that feedback inform an intimate understanding of what users need from the product”. In this regard, it is mentioned the importance of talking to customers through interviews or collecting software usage data. For **business** aspects, the PM is responsible for *owning the product vision*, i.e., “someone who will guide and shape the product to better suit the direction that [...] the founder wants the company to go” (B1). The PM



should *balance user and business*, i.e., define the features depending not only on what customers and users want but what is most interesting for the business. Other related duties related to the business could be *positioning within and monitoring competition*, and *sales and pricing*.

#### 4.3. Knowledge and background

As I5 mentioned: *“the product manager does not have a school”* so we observed that the background of the PMs vary, generally based on one of the areas connected by the role: *software development*, *business*, or *design/user experience*. It is interesting to notice that the interviewees’ and blog post authors’ opinions vary regarding which background is more adequate to the role, generally tending to the background of that particular person. PMs with a *technical background* stress that a better understanding of technical aspects helps *“to realize what goals or features are feasible and what are not”* (B2) and *“understanding the technology behind your product which will not only help in supporting your team of developers or engineers, but will also help with communicating with other parts of your business, and investors”* (B2). A transition from business-related roles to product management is also a natural one. In this regard, we can mention blog post authors mentioning, for instance, a MBA degree (B3) or their experience as a management consultant (B19). Finally, in fewer cases in our sample, it is possible for a designer to become a PM. I5 believes that *“it is the best transition possible”* because of their *“background”* on *“discovery”*.

Although a PM has a background on a specific area, it is clear the need for a *broad knowledge*. Most of the analyzed sources mentioned different degrees of knowledge on the other areas: software development, business, and design/user experience. This capacity is useful not only for performing duties related to these areas as described in the previous section but also to facilitate communication with the other teams. As I4 mentioned: *“I think that it is really important to have a systemic view of all fronts to not be lost in conversations”*. Since PMs generally come from other areas, they must be *continuously learning*, either on the job or through courses or talking to other professionals. In this regard, some excerpts from the blog posts are: *“although they may not have 100% of the skills and knowledge at the famous intersection between usability/design and business and technology, this can be developed over time”* (B4); *“During the past year, I spent my free time learning more about product marketing, SEO, data analytics, gamification, and data science as growth was a priority for my product”* (B5); *“I read books, blogs, listen to podcasts and go to all the meetups”* (B25).

#### 4.4. Dependencies

As mentioned earlier, PMs’ duties and responsibilities vary in certain degree given not only the background but also on other two major aspects: the company or product **stage** and the **context**.

##### 4.4.1. Stage

The stage of the company or the product is a major force dictating not only the role of the PM but even its existence. In the data, it was possible to observe that the PM role is generally not present in the beginning of startups. At this stage, founders generally perform the duties regarding the product vision. As the author of B7 mentioned: *“During my years as a startup founder and CEO, many times I found myself exhausted; I was wearing multiple hats, just like any other early-stage startup founder, by the way. What I didn’t know was that I was also playing the Product Manager role, along with everything else [...]”*

However, the need for a specific person focused on this role becomes essential as the startup grows. At early stages, roles are ill-defined and mingle together. Besides that, the number of features and also customers is smaller. When they grow and the roles become more specialized, people start to flock together in silos focused on specific duties such as software development, user experience, and business development. At this moment, a PM generally is added to the team

to orchestrate the different areas. The PM then acts as a connector between the areas, gluing them together as mentioned in B3: *“it’s only when the team starts to grow beyond 10–15 individuals and people start to specialize in certain functions that communication gaps begin to appear”*. Fig. 4 illustrates this evolution.

##### 4.4.2. Context

Besides the stage, the specific tasks performed by PMs depend on the startup’s context, such as market and team composition. B16 mentions *“we are also in the healthcare industry, where the margin for error is slim”*. I2 compared his previous experience with the startup in which he was working at the moment of the interview: *“[the current startup] sees the product as a way to grow the company. Meanwhile [the previous startup] saw the product as maintenance: I need to have it, otherwise I cannot create my content. But what I really need to do is content”*. In this regard, we should also mention the relevance of technology. For example, B4 mentioned using a framework of tools *“defined by the limitations and scope of the business, the technology, [and customers’ needs]”*.

#### 4.5. Challenges

We identified several challenges mentioned in the sources. Given their granularity, we divided them in the following categories to facilitate the description: **context**, **role definition**, **knowledge**, **product**, and **teamwork**. **Context** refers to aspects of the startup situation, including *time and resources constraints* and *the lack of established processes*. These issues require a pro-active attitude from the PM, as B3 mentions: *“there is always a resource constraint that one is battling with; be it time, money, human resources, or even user behavior data. Instead of waiting for the organization to reach a certain maturity [...], figure out ways to get things done with whatever is available at moment”*. Besides that, *pressure* either from investors or from founders have been mentioned in B30 and by I3.

Challenges about the **role definition** are related to the multidisciplinary nature and lack of a clear definition of what the PM role is. The role is considered a *blend of art and science* as mentioned in B13: *“product management is more art than science, which is where problem with attempting to define it begin”*. Given this lack of a proper definition and background of the PM in a specific area, there is a *risk of focusing on the expertise*, neglecting other areas. As the author of B27 puts: *“you have to remember no matter what role you made this transition from, you have left that job behind you. If you were a developer, designer, or founder, it no longer applies to you”*.

The broad **knowledge** required for the PM is associated with some challenges in this aspect. Sources mention *lack of knowledge on other fields*, *different languages to communicate*, and *no access to experts or mentors*.

Most of the reported challenges are related to the responsibilities regarding the **product**. A key aspect in the context of startups mentioned in the sources was *changing specifications and roadmap* and the *difficulty to balance all stakeholders*. B5 mentions: *“I’ve also been a part of startup teams whose priorities have changed overnight based on an announcement made by their partners or competitors”*. The difficulty in balancing the stakeholders is associated with the role of a bridge that the PM has: *“It’s also not uncommon to receive multiple feature requests from both external (end-users) and internal (sales, marketing, and client success teams) stakeholders, with everyone expecting their feature request to be of the highest priority”* (B3).

Finally, some identified challenges are related to **teamwork**. In this regard, the most common challenge is related to the *lack of authority over others*. As a bridge role, the PM is not an authority in either of the areas being connected. Therefore, they must earn authority and lead *“by influence”*. Some excerpts in the sources in this regard are: *“product managers ‘earn’ their authority and lead mostly by influence”* (B4) and *“PMs have no direct power; they win people over with reason and results. PMs have no direct power, meaning engineers do not report to them”* (B13).

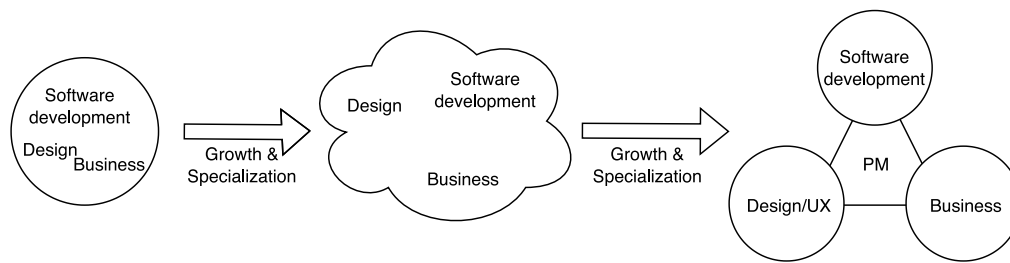


Fig. 4. The emergence of the product manager in software startups as a bridging element.

#### 4.6. Consequences

Although not extensively discussed in the sources, there were some observations regarding the consequences of the existence of a PM in a startup. First of all, these actors have a strong influence on the direction the company takes; in other words, they *shape the future of the startup*. This influence is a direct consequence of the weight the PM has on decisions regarding the product and its vision as discussed in the PM's responsibilities. The fact that, for software startups, this weight manifests in the beginning of the history of the company and the product means that the effects will be felt throughout the history of the company. As the author of one of the blog posts mentioned: *"Being the first product manager is challenging but it also means that you will have a tremendous opportunity to shape the future of the company"* (B10).

Other consequences are related to the role of the PM as a connector between different areas and acting as a middle-man in communication. The existence of a communication broker where initially a direct communication existed introduces a complexity layer and a possible source of friction and issues for the communication. This issue was prevalently expressed in the communication between the software development team and the business areas of the startup. First, *developers have less control over features* as the author of one of the blog posts mentioned: *"the addition of product managers can feel disempowering: developers suddenly have less control over feature definition and priority"* (B18). Then, *stakeholders have less access to developers*. This issue could also lead to frustration because, as written in B18, developers are those who solve the stakeholders' problems.

#### 5. Discussion

The contribution of this paper is a Socio-Technical Grounded Theory concerning PMs' role in software startups, based on data consisting of 30 blog posts and 10 interviews. In this section, we first summarize the answer to the research question, and then we discuss the implications of the results to theory and practice.

##### 5.1. Answer to the research question

Our results show that the PM is an ill-defined role whose responsibilities vary depending on the startup and product context and stage. Although the specific duties vary, this role is multidisciplinary with the tasks regarding, on a functional level, three areas: software development, business development, and design and user experience. Besides these tasks traditionally associated with the areas, the PM is also an agglutinating actor, being responsible for facilitating communication both internally, among the different areas, and externally, with users, customers, and other stakeholders, acting as a bridge. In this regard, PMs also become leaders, shaping the product vision and roadmap and guiding the startup to achieve this vision. They also generally perform managerial tasks, such as coordinating the developing team and facilitating teams' work. These results stress the socio-technical nature of the product management phenomenon by showing how human characteristics, such as knowledge and skills, influence technical aspects, such as the tasks performed, which are, in turn, not only technical but also

contemplate social ones such as leadership and communication. Our results are in line with previous studies on PMs [30,31] and on the role of product owners in large distributed enterprises [56], including the idea of a tailored PM role that has already been discussed by Bass and Haxby [38]. In this regard, our study consolidates these results in an overall theory and extends them to the context of software startups. Specifically, Maglyas et al. [30] had hypothesized that the responsibilities depended on the organizational structure and the top management. By investigating startups in different stages and operating in different contexts, we could observe the influence of these aspects, stages, and contexts on the responsibilities performed by the PM. Examples in this regard are regulated market segments, such as healthcare, that should follow strict guidelines and different levels of importance of the software to the company's strategy. Besides that, since in software startups, PMs quite often migrate from other areas, it was possible to observe the influence of their backgrounds on the way they perform the role. Finally, our results show that PMs in software startups change as the organizations mature and grow, becoming a more specialized and institutionalized role. Although the idea of a PM as a connecting role has already been observed [35], our study describes the emergence of this need in a growing company. As startups grow and their team members specialize in specific roles rather than accumulating functions as in the early stages [57], the need for connecting actors emerges. Besides that, our results support this idea through a study specifically focused on PMs. Maglyas et al. [30] had hypothesized that the emergence of the PM was related to an increase in tasks to be handled by the top management. In our study, we could observe that this phenomenon is actually related to the specialization of people and the increasing need for communication and coordination among the areas that are being formed as the company grows and matures. Regarding product owners, Bass [56] identified nine functions of a product owner team in this context: groom, prioritizer, release master, technical architect, governor, communicator, traveler, intermediary, and risk assessor. However, our theory stresses the importance of the user experience and design aspects, identifying other activities, and relating the tasks to the different areas connected by the PM.

Regarding software development, it is clear the influence of the PM on requirements-related duties, namely, defining the requirements, creating specifications, prioritizing, and validating, corroborating previous results [16] pointing to the PM's influence on RE activities in software startups. Another clear aspect is the influence of other aspects in RE, especially business. Rather than separated concerns as generally described, business and technological decisions are mingled, requiring specific practices, as proposed in the Entrepreneurial Software Engineering Model [58]. This aspect is related to the increasing research on experimentation in software engineering [59–62], especially in software startups [8,63] given the uncertainty regarding users' needs and market [4–6].

In a recent study on the requirements engineer role, Franch et al. [64] performed an interview study and concluded that the role is "elusive", i.e., "hard to comprehend or define". They observed that the role is fulfilled by different positions, such as business or system analyst, engineer, or developer. However, they also mention a PM. Besides requirement-related tasks, they reported other duties, such

as “talking to other groups”, “quick studies or pre-studies” to select requirements for market-driven companies, or performing “go vs. no-go decisions”, in summary, responsibilities similar to those observed for PMs. In this regard, our study points towards an absorption of requirements engineer roles by PMs in the context of software startups or an evolution of such roles. A close relationship between requirements and business aspects of software products is not a new concern [65,66]. Other studies pointed out the challenges of current RE practices regarding aspects that our results related to PMs, e.g., communication [67] and creativity [68]. Finally, current RE practices still struggle to gather feedback from a large crowd [69], or market-driven contexts [70], the general case of software startups that develop software products for a large pool of users.

The specific duties of PMs are also related to the context of the startup and the product stage. This effect has been observed before regarding the RE activities [14] and other software engineering activities, such as testing and project management [10]. It is also interesting that, quite often, at early stages, a founder generally acts as the PM. This phenomenon is related to the founder’s dominance on the actions and competencies in startups [57]. However, while Magliyas et al. [30] observed that PMs generally complain about lack of authority and access to resources, in our study, we observed that some PMs had higher levels of power, even defining sales and pricing strategies. This difference might be related to the small size but also to the speed needed at the initial stages of the company.

It should be pointed out the importance of PMs for communication both inside the team and towards other teams and users. Research has shown that these aspects are key human issues in software engineering [71,72]. In a survey with 192 respondents, Hoffmann et al. [72] assessed the frequency and criticality of human-related challenges in software engineering teams, dividing them into “within teams” and “between teams and clients”. The first category started with “insufficient analysis at the beginning of a task”, “lack of leadership”, and “missing project documentation”. Besides that, at the fifth position, the problem was “information is not made known to the team”, at the seventh, “misunderstandings in communication” followed by “communication plan is neglected”. Regarding client-related challenges, the list started with “lack of interest in the project by the client”, “client does not know what they want”, “no direct communication with the client”. In summary, among other duties, PMs in startups tackle key human-related challenges in software engineering.

Regarding skills and knowledge required for the role, our results show that PMs in software startups have a background in one of the areas connected by the role, i.e., software development, business, or design. It is important to emphasize that most of the PMs interviewed had their first experiences in the role in software startups, generally migrating from other functions, leading to their focus on tasks related to their own expertise and, to some extent, neglecting or under-performing on tasks related to the other areas. However, they must have a broad knowledge encompassing the other areas. These abilities are also needed to enable the person to act as a connector between areas, possessing vocabulary and understanding to communicate effectively. Using the bridge metaphor, PMs build the connection starting from the area they are more used to. Consequently, the bridge is probably “wider” from that side. This variance could explain why the PM role is usually considered “ill-defined”. Another explanation for this issue might be that, as a connecting role, the PM generally “does not produce” a tangible artifact. Besides that, the boundaries between the PM role and the other areas are generally blurred, i.e., it is hard to tell when the areas’ duties end and those from the PM start. This aspect of the background of the PMs has not been investigated in previous studies. It is particularly interesting to observe in software startups that, facing a context of restricted resources, they often have to train a professional for the PM role.

## 5.2. Implications

Our results bring several implications to research, practice, and education in software engineering. Regarding research, the study presents evidence of “growing pains” in software startups as they mature and the emergence of roles related to them. The results also present evidence of the intertwined nature between software engineering, especially regarding requirements, and business development. In general, there is a blurred boundary between software and business development. As B21 mentions: “the value of a product is in what gets used, not simply what gets built”. In this regard, literature has discussed other dimensions of software development success, such as the impact on stakeholders and market performance [73], including the idea of pivots in software startups [7]. A pivot is a strategic change of a business concept, product, or the different elements of a business model [7]. Jantunen and Gause [33] developed a grounded theory on software product management challenges. They concluded that what they called the expert paradigm, where developers are “experts in technologies, tools, software development methods, and project management”, lost effectiveness when the software design problem became more of a wicked problem.

Regarding training and education, it is clear from the sources that PMs with a technical background struggle with deficiencies in knowledge and experience, especially in business. In this regard, it would be interesting to allow closer contact of software engineering students to business topics. Courses on the Lean Startup and Customer Development methods, e.g. [74–76], are good starting points. It is interesting, though, that in this context, a tendency to focus on developing the solutions has been observed [77]. The problem of a deficiency in communication skills for requirements engineers has also been reported [78].

For practitioners, our results present a reference to what is expected from PMs in software startups, especially given the fact that, as mentioned by several of them, there is not formal program or course to become a PM. Based on this reference, PMs could identify their weaknesses and work to improve them both from a knowledge perspective and also from how they behave in the team. It could also help a professional in the transition to becoming a PM. For example, a developer shifting to a PM role could use our results to understand how to prepare herself and what could be expected from her.

In this regard, our results describe some differences between technical leaders, project managers, and product owners. An actor leading the development team, using titles such as tech leader, Chief Technology Officer (CTO), or even scrum master [79], has the responsibility of coordinating and managing the development team, defining people responsible for specific tasks, and taking technical decisions. In some startups, the PM accumulates these functions. Using the reference of the areas as in Fig. 4, the tech leader is generally located in the software development area of the startup.

Project managers are generally associated with an enterprise with a defined scope and time to completion, i.e., as I2 puts “a project is beginning, middle, and end. You have that scope, you start that scope, execute and deliver it, and go to the next”. This context is generally uncommon in startups developing a novel product for a market with many users [6,70]. Hence, PMs have a larger control of the developed features but have to handle endless backlogs and other challenges. I2 concludes: “[For the product,] you sell, create the features, check if it worked, if it hadn’t worked, you go back, if it had, you go to the next. You are watching a metric, you are watching revenue, [...] Of course, every product has an end cycle, it happens [...], but you are not waiting to deliver the next feature or which will be your next project”. In agile methods, the project manager has already been seen as a connector between the project team and stakeholders [80].

The product owner is a role present in agile methodologies, and more specifically, in Scrum. In startups, this role is often performed by the PM. However, as our results show, the role of a PM is much



more comprehensive than that of a product owner even though this role's responsibilities also vary depending on the context [37] including communication [39]. In this regard, we corroborate the argument of Kittlaus [34]: “the spectrum of activities and responsibilities of a product manager is much larger than [...] the product owner role [in agile methods]”.

## 6. Threats to validity

In this section, we discuss the potential threats to the validity of our study and how we mitigated them. To this aim, we followed the guidelines by Wohlin et al. [81] and analyzed the threats to external validity, internal validity, construct validity, and reliability.

**External validity.** This aspect regards to which extent our results are generalizable, being valid outside of the analyzed startups. Since we recruited PMs with different levels of expertise and from different domains, our findings may not necessarily apply to other startups. Although we do not anticipate big differences, additional research is necessary to investigate the transferability of the results. Besides that, we acknowledge that, although we conducted interviews with PMs from startups in different company stages, domains, and countries, we likely did not discover all possible responsibilities, knowledge, and challenges. We are aware that each PM has its singularities and that the startup context is expansive. Our strategies to keep collecting data until reaching information saturation to consider different practitioner profiles and identify recurrent mentions may mitigate this issue.

**Internal validity.** Threats in this category are related to causal inferences and the existence of factors not considered in the study that might be the real causes rather than those identified in the study. Using data from blog posts and interviews, combined with constant comparison, reduced this threat. In this regard, we continued recruiting participants and conducting interviews until we agreed that no new significant information was found. As posed by Corbin and Strauss [45], sampling may be discontinued once the collected data is considered sufficiently dense and data collection no longer generates new information. In our case, interview data was also confronted with the blog information collected.

**Construct validity.** This aspect concerns to what extent the constructs analyzed represent what the researcher had in mind. In qualitative studies, it generally concerns if the terms used have the same meaning for the researchers and the subjects of the study. A potential threat to this aspect of our study regards the use of blog posts whose authors we do not have contact with. We mitigated this issue by comparing several blog posts and interview data in which we could ask for clarifications.

**Reliability.** This aspect regards to what extent the results obtained are dependent on the researchers who performed the study. Following Hoda's Socio-Technical Grounded Theory guidelines [3], we shared evidence of the underlying raw data, the data analysis procedures, and research artifacts to establish both credibility and rigor of application. We also employed a constant comparison method [82] during the data analysis. In this method, each interpretation is constantly compared with existing findings as it emerges from the qualitative analysis.

## 7. Conclusions

Software development is a socio-technical process involving not only technical challenges but also human and business aspects. The comprehension of these challenges boosted the presence of the PM in software development companies. In this paper, we investigated this role in the context of software startups and emerging companies creating novel software-intensive solutions. We were motivated by the uncertain and resource-constrained context of these companies that could intensify the challenges faced by PMs. We performed a Socio-Technical Grounded Theory study based on blog posts and interviews. Our results describe the PM as an ill-defined, multidisciplinary role

focused not only on technical duties but also on communication, strategy, and leadership. This study represents implications for research, practice, and the education and training of software engineers.

The results open several possibilities for future studies. First, the emergence of the PM as an agglutinating factor in a growing startup hints towards the emergence and evolution of roles during the evolution of a startup. Further studies could delve into this aspect. Second, larger studies could better describe the choice of tasks to be performed by a PM. For instance, a study could investigate if there is a minimum set of practices related to software development that always, or in most cases, are performed by the PM. An interesting problem would be to investigate if and how the different dependencies acting in the startup phase might influence the emergence of PMs of the diverse types described in the literature. Another disrupting effect in how PMs perform their activities is the emergence of more powerful AI-based tools, such as a ChatGPT. We collected the data for this study before the announcement of ChatGPT and we expect that it and other similar tools are being employed by product managers. Future work could also consider analyzing other developer-produced artefacts, such as videos or podcasts. Third, the need for improving the education and training of software engineers to consider other aspects, such as communication and business, reinforces the call for research and development of approaches to reach this goal. In this regard, our results could guide the development of guidelines for PMs in software startups. Finally, our study did not identify any difference based on the geographical context of the studied startups. This aspect is worth studying in more detail using our theory to analyze how the different aspects of PMs change in different geographical locations and could be a valuable contribution to future studies.

## CRediT authorship contribution statement

**Jorge Melegati:** Conceptualization, Data curation, Investigation, Methodology, Writing – original draft, Writing – review & editing, Resources. **Igor Wiese:** Conceptualization, Data curation, Investigation, Methodology, Writing – original draft, Writing – review & editing, Resources. **Eduardo Guerra:** Conceptualization, Data curation, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Rafael Chanin:** Investigation, Resources. **Abdullah Aldaej:** Data curation, Investigation, Resources, Visualization. **Tommi Mikko-nen:** Investigation, Resources. **Rafael Prikladnicki:** Investigation, Resources. **Xiaofeng Wang:** Data curation, Investigation, Methodology, Writing – review & editing.

## Declaration of competing interest

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.

## Data availability

Part of the data is shared on Zenodo. Another part is confidential.

## Appendix A. Blog posts

- [B1] Kyle Racki. How to Deploy Product Management in Your Startup. <https://medium.com/@kyleracki/how-to-deploy-product-management-in-your-startup-7a21dc993287>
- [B2] Justus Brown. The ultimate product management cheat sheet for startup CEOs. <https://blog.usejournal.com/the-ultimate-product-management-cheat-sheet-for-startup-ceos-7ac9c452fd7c>
- [B3] Akshat Khanna. My Learnings As The First Product Manager at a Startup. <https://www.linkedin.com/pulse/my-learnings-first-product-manager-startup-akshat-khanna>



- [B4] Andrés Fajardo. Founders and Product Management: A startup story. <https://theventure.city/blog/2019/12/17/founders-and-product-management-a-startup-story/>
- [B5] Bijan. How is Product Management in a Tech Startup Different Than a Large Company?. <https://hackernoon.com/how-is-product-management-in-a-tech-startup-different-than-a-large-company-6gas32fi>
- [B6] Bastiane Huang. My 3 Biggest Takeaways — Being a Product Manager at an Early Stage Startup. <https://productcoalition.com/my-biggest-takeaways-being-a-product-manager-at-an-early-stage-startup-d3122fc5080e>
- [B7] Adriana T. Torresan. Why Every Startup Needs Product Management. <https://www.linkedin.com/pulse/why-every-startup-needs-product-management-adriana-t-torresan>
- [B8] Nithya Hariharan. Should You Be a Startup Product Manager?. <https://productcoalition.com/should-you-be-a-startup-pm-98f994092dc8>
- [B9] Ketan Nayak. The Difference: Being a PM at a Small company vs. a Large Company. <https://productschool.com/blog/product-management-2/difference-pm-small-company-startup-vs-large/>
- [B10] Bastiane Huang. How to be a product manager at an early-stage startup. <https://almanac.io/docs/article-how-to-be-a-product-manager-at-an-early-stage-startup-253ba0231f87c36556eb44748fee812b>
- [B11] Aris Pattakos. Product Management lessons from an Early Stage SaaS Startup. <https://medium.com/swlh/product-management-lessons-from-an-early-stage-saas-startup-4b7e45871d2c>
- [B12] Henry Rahr. How Product Management Scales from Startup to Corporation. <https://www.linkedin.com/pulse/how-product-management-scales-from-startup-corporation-henry-rahr>
- [B13] Kristen Stone. Do Protocols need Product Management?. <https://medium.com/swlh/do-protocols-need-product-management-6c597d65faad>
- [B14] Lenny Rachitsky. This Week #14: Transitioning from startup founder to product manager. <https://www.lennysnewsletter.com/p/this-week-14-transitioning-from-startup>
- [B15] Melissa Perri. Product Manager vs. Product Owner. <https://medium.com/@melissaperri/product-manager-vs-product-owner-57f829aa74d>
- [B16] Hannan Yang. Tips for Founders at their First Product Manager Jobs. <https://medium.com/swlh/tips-for-startup-founders-who-just-started-their-first-product-manager-jobs-49d8ede61427>
- [B17] Hansel Wong. Moving From Product Design to Product Management. <https://medium.com/swlh/moving-from-product-design-to-product-management-ed06ec6a1582>
- [B18] Isaac Souweine. Bringing Product Management to a Scaling Startup. <http://www.isouweine.com/bringing-product-management-to-a-scaling-startup/>
- [B19] Andy Ayim. What is a Product Manager and Why Does My Startup Need One?. <https://www.blacksintechology.net/product-manager-startup-need-one/>
- [B20] WSC Sports. Behind the Scenes: What it Takes to be a Product Manager in a Sports-Tech Startup. <https://medium.com/wsc-sports/behind-the-scenes-what-it-takes-to-be-a-product-manager-in-a-sports-tech-startup-94f3ec99f47>
- [B21] Shipra Mahindra. The difference between being a good product manager and a great one. <https://product.canva.com/startup-product-management/>
- [B22] John Ellison. What Does a Product Manager Do at a Silicon Valley Tech Startup?. <http://writings.john-ellison.com/what-does-a-product-manager-do-at-a-silicon-valley-tech-startup/>
- [B23] Richard Banfield. Product Management Advice for Startup Leaders I Wish Someone Had Told Me. <https://medium.com/agileinsider/product-management-advice-for-startup-leaders-i-wish-someone-had-told-me-2ea121a75ad7>
- [B24] Andrea Saez. Does Your Startup Need a Dedicated Product Manager?. <https://dreasaez.medium.com/does-your-startup-need-a-dedicated-product-manager-d7df2c8cdc0>
- [B25] Greig Cranfield. Product Management at a startup. <https://medium.com/startup-foundation-stories/product-management-at-a-startup-a5758f029cd5>
- [B26] Lenny Rachitsky. What Buddhism Taught Me About Product Management. <https://medium.com/swlh/what-buddhism-taught-me-about-product-management-f05c7486649c>
- [B27] Vindhya C. 15 Take Aways From Being a Product Manager at a High Growth Startup. <https://medium.com/agileinsider/10-lessons-i-learned-being-a-pm-at-a-high-growth-startup-99f7e1cdeecf>
- [B28] Andrew Quan. 4 productive habits for a startup product manager. <https://productcoalition.com/so-youre-a-product-manager-in-a-startup-how-hard-could-it-be-d034f999881a>
- [B29] Jennelle Nystrom. Test-Driven Product Management at an Early-Stage Startup. <https://amplitude.com/blog/test-driven-product-management>
- [B30] Claire Willett. How You Like Dem Hats? Product Management in a Small Startup. <https://www.americaninno.com/boston/how-you-like-dem-hats-product-management-in-a-small-startup/>

## Appendix B. Interview guide

All the interviews followed a semi-structured script and were conducted using a video platform (Zoom). Semi-structured interviews encourage interviewees to freely share their thoughts and enable researchers to follow up and explore interesting topics that might emerge. We recorded each interview and transcribed them. The interviews were conducted following the guide presented below. The script was validated through a pilot interview which was included in the final dataset.

### Questions about the individual

1. Do you have an academic degree? In which area?
2. What have you worked on before joining this company? What were your roles? For how long?
3. Have you worked as a product manager in other companies?
4. Did you have experience with business/technology/UX or UI design (ask about the topics not mentioned before)?
5. Could you mention at least three characteristics from you that you believe are important for your role at the company?

### Questions about the company

6. What is the product your company develops about?
7. Who are the (potential) customers and users?
8. For how long does the company exist?
9. What is the current stage of the company? Only an idea/ developing an initial version/ few customers or users/ mature stage?
10. What is the size of the company? And the team? What is the composition of the team(s)?

### Questions about the role

11. What are your duties and responsibilities as product manager? (Business, technical, and design)
12. What are the most important duties and responsibilities?
13. What is your daily routine?
14. Do you have weekly or monthly duties that are not included in your daily routine?
15. Does the team follow a defined development process? How was this process defined?
16. Do you have management duties?
17. What are the tools you use?

18. What are the challenges you face?
19. Have you worked in other companies as a product manager?  
If yes, what are the differences and similarities to your current role? What were the other companies about? Stage, product, etc.
20. What are the differences between your role and others at your current company or at your previous jobs, such as project manager, technical leader, designer, etc?

#### Concluding questions and feedback

21. Is there any other information you would like to add about your job?
22. Comments and feedback?

## References

- [1] B.H. Cheng, J.M. Atlee, Research directions in requirements engineering, in: *Future of Software Engineering, FOSE'07*, IEEE, 2007, pp. 285–303, <http://dx.doi.org/10.1109/FOSE.2007.17>.
- [2] E. Chisa, Evolution of the product manager, *Commun. ACM* 57 (11) (2014) 48–52, <http://dx.doi.org/10.1145/2669480>.
- [3] R. Hoda, Socio-technical grounded theory for software engineering, *IEEE Trans. Softw. Eng.* (2021) 1–25, <http://dx.doi.org/10.1109/TSE.2021.3106280>.
- [4] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, P. Abrahamsson, Software development in startup companies: A systematic mapping study, *Inf. Softw. Technol.* 56 (10) (2014) 1200–1218, <http://dx.doi.org/10.1016/j.infsof.2014.04.014>.
- [5] V. Berg, J. Birkeland, A. Nguyen-Duc, I.O. Pappas, L. Jaccheri, Software startup engineering: A systematic mapping study, *J. Syst. Softw.* 144 (2018) 255–274, <http://dx.doi.org/10.1016/j.jss.2018.06.043>.
- [6] M. Unterkalmsteiner, P. Abrahamsson, A. Nguyen-duc, G.H. Baltes, K. Conboy, D. Dennehy, R. Sweetman, H. Edison, S. Shahid, X. Wang, J. Garbajosa, T. Gorschek, L. Hokkanen, I. Lunesu, M. Marchesi, L. Morgan, C. Selig, M. Oivo, S. Shah, F. Kon, Software startups - A research agenda, *e-Inf. Softw. Eng. J.* 10 (1) (2016) 1–28, <http://dx.doi.org/10.5277/e-Inf160105>.
- [7] S.S. Bajwa, X. Wang, A. Nguyen Duc, P. Abrahamsson, “Failures” to be celebrated: an analysis of major pivots of software startups, *Empir. Softw. Eng.* 22 (5) (2017) 2373–2408, <http://dx.doi.org/10.1007/s10664-016-9458-0>.
- [8] J. Melegati, H. Edison, X. Wang, XPro: A model to explain the limited adoption and implementation of experimentation in software startups, *IEEE Trans. Softw. Eng.* 48 (6) (2022) 1929–1946, <http://dx.doi.org/10.1109/TSE.2020.3042610>.
- [9] C. Giardino, N. Paternoster, M. Unterkalmsteiner, T. Gorschek, P. Abrahamsson, Software development in startup companies: The greenfield startup model, *IEEE Trans. Softw. Eng.* 42 (6) (2016) 585–604, <http://dx.doi.org/10.1109/TSE.2015.2509970>.
- [10] E. Klotins, M. Unterkalmsteiner, P. Chatzipetrou, T. Gorschek, R. Prikladnicki, N. Tripathi, L.B. Pompermaier, A progression model of software engineering goals, challenges, and practices in start-ups, *IEEE Trans. Softw. Eng.* 47 (3) (2021) 498–521, <http://dx.doi.org/10.1109/TSE.2019.2900213>.
- [11] E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, in: *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Crown Business, 2011.
- [12] S. Blank, *Why the lean start-up changes everything*, *Harv. Bus. Rev.* 91 (5) (2013) 63–72.
- [13] J. Melegati, E. Guerra, X. Wang, Understanding hypotheses engineering in software startups through a gray literature review, *Inf. Softw. Technol.* 133 (2021) 106465, <http://dx.doi.org/10.1016/j.infsof.2020.106465>.
- [14] C. Gralha, D. Damian, A.I.T. Wasserman, M. Goulão, J. Araújo, The evolution of requirements practices in software startups, in: *Proceedings of the 40th International Conference on Software Engineering - ICSE '18*, ACM Press, New York, New York, USA, 2018, pp. 823–833, <http://dx.doi.org/10.1145/3180155.3180158>.
- [15] N. Tripathi, E. Klotins, R. Prikladnicki, M. Oivo, L. Pompermaier, M. Arun Sojan Kudakacheril, A., Unterkalmsteiner, K. Liukkunen, T. Gorschek, An anatomy of requirement engineering in software startups using multi-vocal literature and case survey, *J. Syst. Softw.* 146 (2018) 130–151, <http://dx.doi.org/10.1016/J.JSS.2018.08.059>.
- [16] J. Melegati, A. Goldman, F. Kon, X. Wang, A model of requirements engineering in software startups, *Inf. Softw. Technol.* 109 (2019) 92–107, <http://dx.doi.org/10.1016/j.infsof.2019.02.001>.
- [17] I. De Weerd, S. Brinkkemper, R. Nieuwenhuis, J. Versendaal, L. Bijlsma, On the creation of a reference framework for software product management: Validation and tool support, in: *2006 International Workshop on Software Product Management (IWSPM'06 - RE'06 Workshop)*, IEEE, 2006, pp. 3–12, <http://dx.doi.org/10.1109/IWSPM.2006.6>.
- [18] H.-B. Kittlaus, P.N. Clough, *Software products: Terms and characteristics*, in: *Software Product Management and Pricing*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 5–15, [http://dx.doi.org/10.1007/978-3-540-76987-3\\_2](http://dx.doi.org/10.1007/978-3-540-76987-3_2).
- [19] C. Ebert, S. Brinkkemper, *Software product management – An industry evaluation*, *J. Syst. Softw.* 95 (2014) 10–18, <http://dx.doi.org/10.1016/j.jss.2013.12.042>.
- [20] C. Florescu, C. Caragea, PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents, in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1, Association for Computational Linguistics, Stroudsburg, PA, USA, 2017, pp. 1105–1115, <http://dx.doi.org/10.18653/v1/P17-1102>.
- [21] J. Melegati, E. Guerra, I. Scaliante Wiese, X. Wang, Generated abstracts: evaluating automatic text summarization for blog posts in gray literature studies, in: *The International Conference on Evaluation and Assessment in Software Engineering Engineering 2022*, ACM, New York, NY, USA, 2022, pp. 282–287, <http://dx.doi.org/10.1145/3530019.3535308>.
- [22] P. Kotler, *Principles of Marketing : Activebook Version 2.0*, tenth ed., Prentice Hall, Upper Saddle River, NJ [etc.], 2004.
- [23] S.A. Fricker, *Software product management*, in: A. Maedche, A. Botzenhardt, L. Neer (Eds.), *Software for People: Fundamentals, Trends and Best Practices*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 53–81, [http://dx.doi.org/10.1007/978-3-642-31371-4\\_4](http://dx.doi.org/10.1007/978-3-642-31371-4_4).
- [24] K. Beck, C. Andres, *Extreme programming explained: Embrace change*, second ed., XP Series, Pearson Education, 2004.
- [25] H. Kittlaus, *Software Product Management: The ISPM®-Compliant Study Guide and Handbook*, Springer Berlin Heidelberg, 2022.
- [26] T. DeMarco, T. Lister, *Software development: State of the art vs. State of the practice*, in: *11th International Conference on Software Engineering, IEEE*, 1989, pp. 271–275, <http://dx.doi.org/10.1109/ICSE.1989.714432>, URL <http://ieeexplore.ieee.org/document/714432/>.
- [27] C. Neill, P. Laplante, *Requirements engineering: the state of the practice*, *IEEE Softw.* 20 (6) (2003) <http://dx.doi.org/10.1109/MS.2003.1241365>.
- [28] P. Kruchten, R. Capilla, J.C. Duenas, The decision view's role in software architecture practice, *IEEE Softw.* 26 (2) (2009) 36–42, <http://dx.doi.org/10.1109/MS.2009.52>.
- [29] E. Klotins, M. Unterkalmsteiner, T. Gorschek, *Software engineering in start-up companies: An analysis of 88 experience reports*, *Empir. Softw. Eng.* 24 (1) (2019) 68–102, <http://dx.doi.org/10.1007/s10664-018-9620-y>.
- [30] A. Maglyas, U. Nikula, K. Smolander, What are the roles of software product managers? An empirical investigation, *J. Syst. Softw.* 86 (12) (2013) 3071–3090, <http://dx.doi.org/10.1016/j.jss.2013.07.045>.
- [31] O. Springer, J. Miler, The role of a software product manager in various business environments, in: *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, FedCSIS 2018*, Vol. 15, Polish Information Processing Society, 2018, pp. 985–994, <http://dx.doi.org/10.15439/2018F100>.
- [32] H.-B. Kittlaus, S.A. Fricker, *Software product management: The ISPM-compliant study guide and handbook*, *Management of Computing and Information Systems*, Springer, 2017, p. 292.
- [33] S. Jantunen, D.C. Gause, Using a grounded theory approach for exploring software product management challenges, *J. Syst. Softw.* 95 (2014) 32–51, <http://dx.doi.org/10.1016/j.jss.2014.03.050>.
- [34] H.-B. Kittlaus, *Software product management and agile software development: Conflicts and solutions*, in: A. Maedche, A. Botzenhardt, L. Neer (Eds.), *Software for People: Fundamentals, Trends and Best Practices*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 83–96, [http://dx.doi.org/10.1007/978-3-642-31371-4\\_5](http://dx.doi.org/10.1007/978-3-642-31371-4_5).
- [35] T. Sedano, P. Ralph, C. Peraire, The product backlog, in: *2019 IEEE/ACM 41st International Conference on Software Engineering, ICSE, IEEE*, 2019, pp. 200–211, <http://dx.doi.org/10.1109/ICSE.2019.00036>.
- [36] J. Sutherland, K. Schwaber, *The scrum guide - the definitive guide to scrum: the rules of the game*, 2017, URL <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>.
- [37] H.S. Sverrisdottir, H.T. Ingason, H.I. Jonasson, The role of the product owner in scrum-comparison between theory and practices, *Procedia - Soc. Behav. Sci.* 119 (2014) 257–267, <http://dx.doi.org/10.1016/j.sbspro.2014.03.030>.
- [38] J.M. Bass, A. Haxby, Tailoring product ownership in large-scale agile projects: Managing scale, distance, and governance, *IEEE Softw.* 36 (2) (2019) 58–63, <http://dx.doi.org/10.1109/MS.2018.2885524>.
- [39] C. Unger-Windeler, J.A.-C. Klünder, T. Reuscher, K. Schneider, Are product owners communicators? A multi-method research approach to provide a more comprehensive picture of Product Owners in practice, *J. Softw.: Evol. Process* 33 (1) (2021) 1–23, <http://dx.doi.org/10.1002/smr.2311>.
- [40] M. Crowne, Why software product startups fail and what to do about it. Evolution of software product development in startup companies, in: *IEEE International Engineering Management Conference*, 2002, pp. 338–343, <http://dx.doi.org/10.1109/IEEM.2002.1038454>.
- [41] J. Melegati, X. Wang, P. Abrahamsson, Hypotheses engineering: First essential steps of experiment-driven software development, in: *2019 IEEE/ACM Joint 4th International Workshop on Rapid Continuous Software Engineering and 1st International Workshop on Data-Driven Decisions, Experimentation and Evolution (RCoSE/DDrEE)*, IEEE, 2019, pp. 16–19, <http://dx.doi.org/10.1109/RCoSE/DDrEE.2019.00011>.

- [42] R.P. Bostrom, J.S. Heinen, MIS problems and failures: A socio-technical perspective. Part I: The causes, *MIS Q.* 1 (3) (1977) 17, <http://dx.doi.org/10.2307/248710>.
- [43] K.-J. Stol, P. Ralph, B. Fitzgerald, Grounded theory in software engineering research, in: *Proceedings of the 38th International Conference on Software Engineering*, ACM, New York, NY, USA, 2016, pp. 120–131, <http://dx.doi.org/10.1145/2884781.2884833>.
- [44] P. Ralph, N. bin Ali, S. Baltes, D. Bianculli, J. Diaz, Y. Dittrich, N. Ernst, M. Felderer, R. Feldt, A. Filieri, B.B.N. de França, C.A. Furia, G. Gay, N. Gold, D. Graziotin, P. He, R. Hoda, N. Juristo, B. Kitchenham, V. Lenarduzzi, J. Martínez, J. Melegati, D. Mendez, T. Menzies, J. Moller, D. Pfahl, R. Robbes, D. Russo, N. Saarimäki, F. Sarro, D. Taibi, J. Siegmund, D. Spinellis, M. Staron, K. Stol, M.-A. Storey, D. Taibi, D. Tamburri, M. Torchiano, C. Treude, B. Turhan, X. Wang, S. Vegas, Empirical standards for software engineering research, 2021, [arXiv:2010.03525](https://arxiv.org/abs/2010.03525).
- [45] J. Corbin, A. Strauss, *Basics of Qualitative Research*, in: *Core textbook*, SAGE Publications, 2015.
- [46] A. Rainer, Using argumentation theory to analyse software practitioners' de-feasible evidence, inference and belief, *Inf. Softw. Technol.* 87 (2017) 62–80, <http://dx.doi.org/10.1016/j.infsof.2017.01.011>.
- [47] A. Rainer, A. Williams, Using blog-like documents to investigate software practice: Benefits, challenges, and research directions, *J. Softw.: Evol. Process* 31 (11) (2019) 1–22, <http://dx.doi.org/10.1002/smr.2197>.
- [48] B.A. Kitchenham, L. Madeyski, D. Budgen, How should software engineering secondary studies include grey material, *IEEE Trans. Softw. Eng.* XX (2022) 0–3, <http://dx.doi.org/10.1109/TSE.2022.3165938>.
- [49] V. Garousi, M.V. Mäntylä, When and what to automate in software testing? A multi-vocal literature review, *Inf. Softw. Technol.* 76 (2016) 92–117, <http://dx.doi.org/10.1016/j.infsof.2016.04.015>.
- [50] M. Aniche, C. Treude, I. Steinmacher, I. Wiese, G. Pinto, M.A. Storey, M.A. Gerosa, How modern news aggregators help development communities shape and share knowledge, in: *Proceedings - International Conference on Software Engineering*, 2018, pp. 499–510, <http://dx.doi.org/10.1145/3180155.3180180>.
- [51] J. Soldani, D.A. Tamburri, W.-J. Van Den Heuvel, The pains and gains of microservices: A systematic grey literature review, *J. Syst. Softw.* 146 (2018) 215–232, <http://dx.doi.org/10.1016/j.jss.2018.09.082>.
- [52] V. Garousi, M. Felderer, M.V. Mäntylä, Guidelines for including grey literature and conducting multivocal literature reviews in software engineering, *Inf. Softw. Technol.* 106 (2019) 101–121, <http://dx.doi.org/10.1016/j.infsof.2018.09.006>.
- [53] F. Kamei, I. Wiese, C. Lima, I. Polato, V. Nepomuceno, W. Ferreira, M. Ribeiro, C. Pena, B. Cartaxo, G. Pinto, S. Soares, Grey literature in software engineering: A critical review, *Inf. Softw. Technol.* 138 (2021) 106609, <http://dx.doi.org/10.1016/j.infsof.2021.106609>.
- [54] S. Baltes, P. Ralph, Sampling in software engineering research: a critical review and guidelines, *Empir. Softw. Eng.* 27 (4) (2022) 94, <http://dx.doi.org/10.1007/s10664-021-10072-8>.
- [55] R. Weiss, *Learning From Strangers: The Art and Method of Qualitative Interview Studies*, Free Press, 1995.
- [56] J.M. Bass, How product owner teams scale agile methods to large distributed enterprises, *Empir. Softw. Eng.* 20 (6) (2015) 1525–1557, <http://dx.doi.org/10.1007/s10664-014-9322-z>.
- [57] P. Seppanen, M. Oivo, K. Liukkunen, The initial team of a software startup Narrow-shouldered innovation and broad-shouldered implementation, in: *2016 International Conference on Engineering, Technology and Innovation/IEEE International Technology Management Conference (ICE/ITMC)*, IEEE, 2016, pp. 1–9, <http://dx.doi.org/10.1109/ICE/ITMC39735.2016.9026055>.
- [58] M. Kuhrmann, J. Muench, J. Klunder, Hacking or engineering? Towards an extended entrepreneurial software engineering model, in: *Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering*, ACM, New York, NY, USA, 2022, pp. 66–76, <http://dx.doi.org/10.1145/3529320.3529328>.
- [59] F. Fagerholm, A. Sanchez Guinea, H. Mäenpää, J. Münch, The RIGHT model for continuous experimentation, *J. Syst. Softw.* 123 (2017) 292–305, <http://dx.doi.org/10.1016/j.jss.2016.03.034>.
- [60] S.G. Yaman, M. Munezero, J. Münch, F. Fagerholm, O. Syd, M. Aaltola, C. Palmu, T. Männistö, Introducing continuous experimentation in large software-intensive product and service organisations, *J. Syst. Softw.* 133 (2017) 195–211, <http://dx.doi.org/10.1016/j.jss.2017.07.009>.
- [61] A. Fabijan, P. Dmitriev, C. McFarland, L. Vermeer, H. Holmström Olsson, J. Bosch, Experimentation growth: Evolving trustworthy A/B testing capabilities in online software companies, *J. Softw.: Evol. Process* 30 (12) (2018) e2113, <http://dx.doi.org/10.1002/smr.2113>.
- [62] F. Auer, R. Ros, L. Kaltenbrunner, P. Runeson, M. Felderer, Controlled experimentation in continuous experimentation: Knowledge and challenges, *Inf. Softw. Technol.* 134 (2021) 106551, <http://dx.doi.org/10.1016/j.infsof.2021.106551>.
- [63] M. Gutbrod, J. Münch, M. Tichy, How do software startups approach experimentation? Empirical results from a qualitative interview study, in: M. Felderer, D. Méndez Fernández, B. Turhan, M. Kalinowski, F. Sarro, D. Winkler (Eds.), *Product-Focused Software Process Improvement*, Springer International Publishing, Cham, 2017, pp. 297–304, [http://dx.doi.org/10.1007/978-3-319-69926-4\\_21](http://dx.doi.org/10.1007/978-3-319-69926-4_21).
- [64] X. Franch, C. Palomares, T. Gorschek, On the requirements engineer role, *Commun. ACM* 64 (6) (2021) 69–75, <http://dx.doi.org/10.1145/3418292>.
- [65] L. Lehtola, M. Kauppinen, J. Vähäniitty, M. Komssi, Linking business and requirements engineering: Is solution planning a missing activity in software product companies? *Requir. Eng.* 14 (2) (2009) 113–128, <http://dx.doi.org/10.1007/s00766-009-0078-8>.
- [66] M. Jarke, P. Loucopoulos, K. Lyytinen, J. Mylopoulos, W. Robinson, The brave new world of design requirements, *Inf. Syst.* 36 (7) (2011) 992–1008, <http://dx.doi.org/10.1016/j.is.2011.04.003>.
- [67] D. Hidellaraachchi, J. Grundy, R. Hoda, K. Madampe, The effects of human aspects on the requirements engineering process: A systematic literature review, *IEEE Trans. Softw. Eng.* 48 (6) (2022) 2105–2127, <http://dx.doi.org/10.1109/TSE.2021.3051898>.
- [68] R. Mohanani, P. Ralph, B. Turhan, V. Mandic, How templated requirements specifications inhibit creativity in software engineering, *IEEE Trans. Softw. Eng.* (2021) 1–13, <http://dx.doi.org/10.1109/TSE.2021.3112503>.
- [69] E.C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini, M. Stade, The crowd in requirements engineering: The landscape and challenges, *IEEE Softw.* 34 (2) (2017) 44–52, <http://dx.doi.org/10.1109/MS.2017.33>.
- [70] C. Palomares, X. Franch, C. Quer, P. Chatzipetrou, L. López, T. Gorschek, The state-of-practice in requirements elicitation: an extended interview study at 12 companies, *Requir. Eng.* 26 (2) (2021) 273–299, <http://dx.doi.org/10.1007/s00766-020-00345-x>.
- [71] P. Lenberg, R. Feldt, L.G. Wallgren, Human factors related challenges in software engineering - An industrial perspective, in: *Proceedings - 8th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2015*, 2015, pp. 43–49, <http://dx.doi.org/10.1109/CHASE.2015.13>.
- [72] M. Hoffmann, D. Mendez, F. Fagerholm, A. Luckhardt, The human side of Software Engineering Teams: an investigation of contemporary challenges, *IEEE Trans. Softw. Eng.* 5589 (c) (2022) 1, <http://dx.doi.org/10.1109/TSE.2022.3148539>.
- [73] P. Ralph, P. Kelly, The dimensions of software engineering success, in: *Proceedings of the 36th International Conference on Software Engineering - ICSE 2014*, ACM Press, New York, New York, USA, 2014, pp. 24–35, <http://dx.doi.org/10.1145/2568225.2568261>.
- [74] E.F.Z. da Cruz, A. Alvaro, Introduction of entrepreneurship and innovation subjects in a computer science course in Brazil, in: *2013 IEEE Frontiers in Education Conference, FIE, IEEE*, 2013, pp. 1881–1887, <http://dx.doi.org/10.1109/FIE.2013.6685162>.
- [75] R. Chanin, A. Sales, L. Pompermaier, R. Prikladnicki, Challenge based startup learning: a framework to teach software startup, in: *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education - ITICSE 2018*, ACM Press, New York, New York, USA, 2018, pp. 266–271, <http://dx.doi.org/10.1145/3197091.3197122>.
- [76] X. Wang, D. Khanna, M. Mondini, Teaching “Through” entrepreneurship: An experience report, in: *Fundamentals of Software Startups*, Springer International Publishing, Cham, 2020, pp. 235–250, [http://dx.doi.org/10.1007/978-3-030-35983-6\\_14](http://dx.doi.org/10.1007/978-3-030-35983-6_14).
- [77] J. Melegati, R. Chanin, X. Wang, A. Sales, R. Prikladnicki, Perceived benefits and challenges of learning startup methodologies for software engineering students, in: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education - SIGCSE '19*, ACM Press, New York, New York, USA, 2019, pp. 204–210, <http://dx.doi.org/10.1145/3287324.3287382>.
- [78] I. Morales-Ramirez, L.H. Alva-Martinez, Requirements analysis skills: How to train practitioners? in: *2018 IEEE 8th International Workshop on Requirements Engineering Education and Training, REET, IEEE*, 2018, pp. 24–29, <http://dx.doi.org/10.1109/REET.2018.00009>.
- [79] S.V. Spiegler, C. Heinecke, S. Wagner, An empirical study on changing leadership in agile teams, *Empir. Softw. Eng.* 26 (3) (2021) 41, <http://dx.doi.org/10.1007/s10664-021-09949-5>.
- [80] Y. Shastri, R. Hoda, R. Amor, The role of the project manager in agile software development projects, *J. Syst. Softw.* 173 (2021) 110871, <http://dx.doi.org/10.1016/j.jss.2020.110871>.
- [81] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, *Planning*, in: *Experimentation in Software Engineering*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 89–116, [http://dx.doi.org/10.1007/978-3-642-29044-2\\_8](http://dx.doi.org/10.1007/978-3-642-29044-2_8).
- [82] B.G. Glaser, A.L. Strauss, *Discovery of Grounded Theory: Strategies for Qualitative Research*, Routledge, 2017.