Bridging the Gap Between Software Product Management and Software Development Product Data Practices

University of Turku
Department of Computing
Bachelor's Thesis
Computer Science
November 2025
Janina Paasila

UNIVERSITY OF TURKU
Department of Computing

Janina Paasila: Bridging the Gap Between Software Product Management and Software Development Product Data Practices

Bachelor's Thesis, 29 p.
Computer Science
November 2025

Software product management and software development are both key functions for creating and managing software products. Still, these functions are largely seen as separate and most frameworks do not include each other, especially in terms of product data. Software products generate vast amounts of complex data that is largely untapped. Effective management and use of product data could create efficiency and improve decision making leading to higher value creation and better products.

This thesis is a literature review aiming to find out the current product data practices in software product management and software development and how the functions could be better aligned in terms of product data. The research finds that in general the data practices in software product management are high level, focusing on decision making and actionable insights, while software development focuses on big data and testing. Both functions suffer from the same challenges related to the collection, management, and especially the effective use of product data. Various emerging practices, mainly machine learning, are proposed to solve these issues and overall improve the management and use of product data. Frameworks integrating the two functions and involving clear product data practices are required as well as data tools that effectively visualise data and create actionable insights.

Keywords: software product management, software development, product data, data practices, data challenges

Contents

List of Figures

List of Tables

1 Introduction

Digitalisation has brought software products all around us as applications, programs, and website services on different devices. A software product is a digital product or service that consists of functionality and features based on requirements set by stakeholders, especially end users, and resources, such as available technology and budget. Software product management (SPM) orchestrates the process of designing, planning, and prioritising these functionality and features based on the requirements, while managing the product overall and communicating with stakeholders. Software development (SD), on the other hand, is the function building the product's functionality and features while considering the technical opportunities and limitations. Thus, software product management and software development are functions operating closely together and this relationship is rarely acknowledged in academia. [1][2][3]

Managing software products requires constant decision making in complex and unpredictable environments. Historically professionals working with software products have had to rely on personal opinions, experience, intuition, domain knowledge, and stakeholders to guide their decision making [4][5][6][7]. However, as collecting and storing data has become more efficient and cheaper, product data exist as a largely untapped potential for significantly improving the quality and accuracy of product-related decisions. Data-driven decisions are better decisions, limiting the effects of misunderstanding, faulty assumptions, and power relations between stakeholders while providing timely and realistic feedback directly from the product and its users [5]. Software developers and software product managers are largely biased in their grounds for decision making, stating that data is the most important factor while mostly basing their decisions on subjective grounds [4].

Despite the existence of vast data there are various challenges related to the utilisation of product data. In a case study by Moe et al. [2] poor data management was found to be the third most prevalent challenge for software product managers, preceded only by workload and prioritisation issues. Thus, there is a need to find solutions that support software product managers in making better informed decisions, further improving the quality of software product management function. Software product management has existed as an established function for a few decades, while software development function has much further reaching roots and is more extensively researched. Hence, data practices proposed in software development could provide answers for the data-related issues in software product management.

This thesis aims to answer the following questions:

- **RQ1:** What is the role of product data in SPM?
- **RQ2:** What are the relevant themes related to product data in SD?
- **RQ3:** How can SPM and SD data practices be better aligned?

To answer these questions a literature review was conducted. To answer RQ1 (Table 1.1), the following search query was used: *("product manage*" OR "product plann*") AND (software OR digital) AND ("product data" OR "analytics" OR "data-driven").* As product management and product planning are used interchangeably in academic literature, both were included. Similarly, literature sometimes discusses just product management when referring to software or digital products, and as such searching for software product management specifically could limit important results. To answer RQ2 (Table 1.2), a similar search query was

Database	**Search field**	**All results**	**Scanned**	**Final**
Web of Science	topic	17	3	0
ACM	abstract	3	0	0
IEEE	all metadata	26	7	2
ScienceDirect	title, abstract, keywords	2	0	0
Scopus	title, abstract, keywords	40	7	0
SpringerLink	keywords	422	9*	3
		total	16	5

Table 1.1: SLR results for RQ1.

Database	**Search field**	**All results**	**Scanned**	**Final**
Web of Science	topic	55	5	1
ACM	abstract	29	4	0
IEEE	all metadata	109	3	1
ScienceDirect	title, abstract, keywords	15	1	0
Scopus	title, abstract, keywords	104	13	1
SpringerLink	keywords	1731	15*	3
		total	20	6

Table 1.2: SLR results for RQ2.

used: *("software develop*" AND (product OR products) AND ("product data" OR "analytics" OR "data-driven").*

Searches were made in six databases including Web of Science, ACM, IEEE, ScienceDirect, Scopus, and SpringerLink. Results were additionally limited by using inclusion criteria of being in English and being published on or after 2020. Database-specific limitations on which field the search query is applied to can be seen in Tables 1.1 and 1.2. As Scopus and SpringerLink searches produced too many results to go through, further inclusion criteria of "content type: articles and conference papers" for SpringerLink and "discipline: CS and Engineering" for both were applied. Additionally, for SpringerLink only the first 300 results were scanned as the number of results was large and after sorting by relevance most results after the first 200 were not relevant. After scanning the title and abstract in the search results and removing duplicates a total of 16 articles for RQ1 and 20 articles for RQ2 were selected for further analysis. Based on further analysis 5 articles were chosen for RQ1 and 6 articles for RQ2 as the final references.

The thesis is structured as follows: Chapter 2 shortly introduces software product management as a practice with a relevant framework and how SPM relates to software development. Chapter 3 discusses data practices related to software product management and software development, their differences and similarities. It also introduces taxonomies of product data, challenges related to product data, as well as emerging practices within product data analytics. Discussion in Chapter 4 discusses the findings, ties the insights to previous research findings and answers the research questions. Finally, chapter 5 concludes the research and presents its limitations and future research needs.

2 Background

This chapter introduces software product management as a function including its responsibilities and presents a relevant framework detailing the function. It also discusses how software product management is related to software development and distinguishes the functions from each other.

Software product management is responsible for the process of creating, delivering, and maintaining a software product, which involves requirement definition, release management, roadmapping, and lifecycle management [1]. A product manager makes various decisions related to managing and prioritising the product, its features, and requirements [6]. Product management (PM) is closely related to and partially overlapping with product portfolio management (PPM), productisation, product lifecycle management (PLM), and product data management (PDM) [8]. Springer & Miler [9] establish software product management as a business-focused area, where the main focus is to ensure economic success and maximum profit via managing the product lifecycle. During its lifecycle a software product is conceived, planned, developed, qualified, launched, delivered, and retired. While software product management function is responsible for the entire lifecycle [3], rarely is a single product manager responsible for the entire lifecycle but rather they focus on set core activities [9].

Software product managers operate in a volatile environment [10]. Value creation is at the core of software product manager's role where through discovering, experimenting with, and delivering features product value is maximised [11]. Product value is evaluated based on the business value the product provides to both the company and its customers [6]. Traditionally, software product managers rely on best guesses, personal opinions, and experience when making product-related decisions, to which data can provide factual support and evidence [1]. Software product's characteristics as well as the surrounding factors such as product maturity, user types, and technology create complex decision environments [6]. The size and complexity of the product company's internal structure makes product-related decision making even more difficult [9].

Software product management is increasingly becoming integrated into software development frameworks [9]. Holmström Olsson & Bosch [1][11] define software product management to be concerned with what to build and why, while software development focuses on how to build. Thus, software product management connects with functions outside of software development [2]. Software development is focused on the architecture, building, implementation, and testing of software, while software product management considers the product as a whole. Software product management is distinguished from software development in that SPM has a distinct product focus while simultaneously considering a combination of value creation, business requirements, and technical perspective [3]. Further, software product management is coined as the link between business and software development bridging strategic and marketing activities with technical perspective [11]. Whether the software product manager is more focused on the business or technical perspective varies between organisations and roles [12].

There are various frameworks used in software product management including ISPMA framework, SAFe, SPM competence model, and SPM reference framework [11]. The numerous frameworks and overall difficulty in defining software product management is caused by the function's novelty and the lack of shared definition neither in industry nor academia [12]. Here the framework by the International Software Product Management Association (ISPMA) [13] was chosen to be presented in detail, as it is the most comprehensive model, up to date, and specific to software products as opposed to physical, embedded or service products. Based on Figure 2.1 it becomes clear that software product management is heavily focused on product strategy and planning, while software development is its own function focused on the technical details. The product strategy tasks include product positioning, delivery model, ecosystem management, sourcing, pricing, financial management, legal management, and risk management. The product planning tasks overarch collecting and evaluating user insights, managing product lifecycle, roadmapping, release planning, and defining requirements. In addition to product strategy and planning, the SPM function is responsible for analysing the product and its market. The SPM function is also responsible for overseeing software development, product marketing, sales, and product support, as well as participating in company-level strategic management.

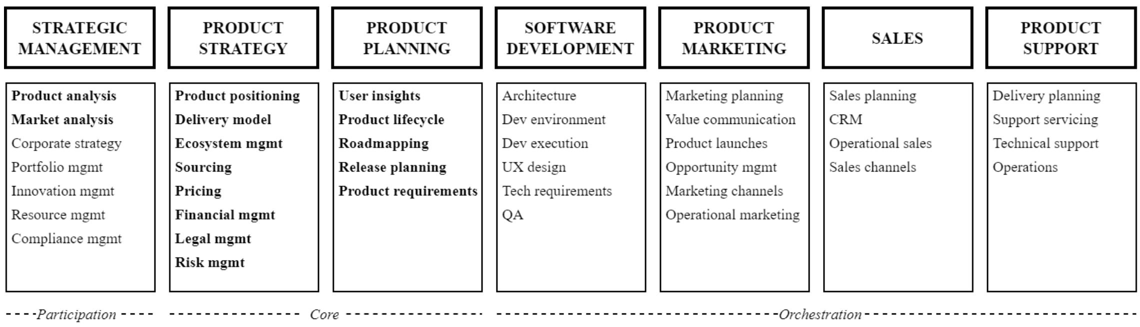

Figure 2.1: ISPMA framework with SPM responsibilities [13].

3 Product Data Practices in Software Product Management and Software Development

Academia has discussed various ways how software product management and software development functions collect, analyse, and use product data. The product data comes in large volumes from a multitude of sources and formats, creating issues with its management and use. Data is considered to be the key for improving software product management, but to achieve such improvement data practices must overcome various challenges, for example by exploiting emerging technologies. This chapter discusses the product data practices in SPM and SD, as presented in Table 3.1, while considering the importance of product data for guiding product decisions.

Table 3.1: Research focus areas related product data practices in SPM and SD.

Data-Driven Product Management (SPM)	**Software Analytics (SD)**
Improving decision making [10][11]	Managing big data [14]
Creating shared product understanding between stakeholders [8]	Data-driven organisation and teams [15]
Procuring actionable insights [10]	A/B-testing [3][16]
Feedback loop effectiveness [8]	Software testing [16]
Evaluating value creation [11]	Evaluating value creation [3]
Requirements definition process [1][11]	Automating requirements definition [14]

Data-driven decision making is argued to produce informed and more reliable decisions [10]. It is coined to be a key in solving the issue of lacking shared understanding of a product between stakeholders [8]. Data-driven product management (DDPM) has emerged as an overarching field where the focus is on systematically procuring actionable insights from vast raw data [10]. The counterpart in software development is software analytics (SA), which includes using data analytics to make informed software-related decisions [15]. A software product generates various data throughout its lifecycle. The data a product generates over its lifetime comes in large volumes and is heterogeneous, also termed big data [14]. The characteristics of the product and its users determine not just data collection, but especially how and which data is used for analytics [8]. The effective use of data provides significant potential for improving prioritisation and decision making [11]. Data-driven organisational culture and infrastructure can further support the effective use of data in developing software products [15].

Product data is used for both developing new features as well as improving existing ones [11]. Receiving continuous feedback from the product via data creates an effective feedback loop between software product management and product users [8]. In most cases product data is extensively available to support and guide development efforts and it plays an important role in evaluating value creation following product decisions [11]. Product data helps to reveal the true needs and behaviours of users and highlight anomalies or unexpected patterns, even when users are unable to voice their needs. However, data usage for product-related decision making and validation is suggested to be complemented with subjective grounds such as experience and input from stakeholders [15].

The role of software product management is especially relevant in defining requirements [3][11], which is a key function for software development as well [17]. Data-driven development and continuous deployment challenge the assumption that requirements are something that can be defined before development [1]. Adopting data-driven ways of working shifts defining requirements into hypothesis testing [11]. Traditionally requirements are gathered from stakeholders via interviews and other qualitative measures [14][17]. Benefiting from data offers an opportunity to elicit requirements from various alternative sources, automate the collection of potential requirements, and increase the frequency of new releases. Thus, there is an urgent need for methods to connect heterogeneous data points and autonomously discover requirements [14].

While requirements are gathered from various data sources and with user research, one of the main techniques for validating requirements, features and related assumptions is A/B-testing, also termed controlled experimentation or continuous experimentation (CE) [3][16]. CE aims to evaluate either the problem-solution or product-market fit [3]. Moreover, A/B-testing is used to complement software testing to gain real feedback from real users [16]. A/B-testing uses two variants of a feature prototype to gain feedback for selecting how a requirement should be realised. The prototypes used may be low-fidelity, such as wireframes or conceptual drawings, or high-fidelity, such as operational demos or fully developed features. Results of A/B-testing are mainly used for feature selection and secondarily for feature rollout, continued feature development, subsequent A/B-testing, validation of the A/B-testing, and validation of the research question. Overall, CE plays an important role in software development for evaluating value creation [3].

3.1 Software Product Data

Software product-related data has been categorised in various ways. The categories include various data points that can be qualitative or quantitative in nature and collected pre- or post-deployment. The categorisations found in the research are presented here by increasing level of detail. Figure 3.1 presents an adapted visual-

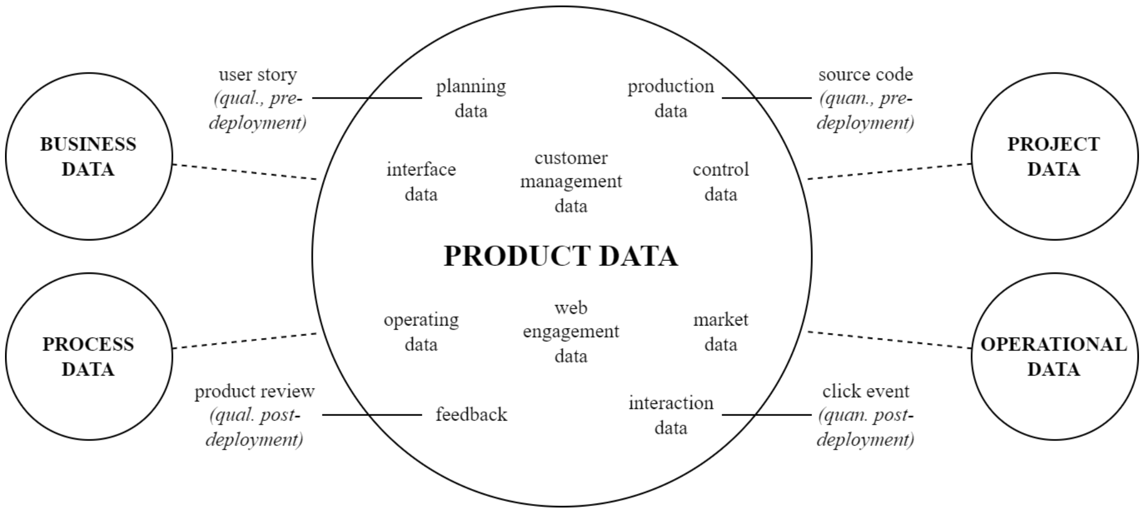

Figure 3.1: Visualisation of software product data based on Fichtler et al. [10] and Fawzy et al. [18].

isation of the taxonomy by Fichtler et al. [10] and related data types by Fawzy et al. [18] with examples of single data points.

Holmström Olsson & Bosch [11] distinguish historical (data describing behaviour with the system) and runtime data (data describing behaviour of the system). Harkonen et al. [8] divide data types into master data (data generated during development), transactional data (data generated via business processes), and interactional data (data generated during product usage). Quin et al. [16] present four data types in relation to A/B-testing, including product/system data (technical data related to the product and user), user-centric data (behavioural data from interaction with the product), spatial-temporal data (location and time data), and secondary data (other uncategorised data). Additionally, Lim et al. [14] discuss dynamic data as a new data source where insights can be gathered from digital unintended sources, such as online reviews and social media comments, that are not purposed for acting as feedback for the product management (see 3.3).

Fichtler et al. [10] presented a detailed data taxonomy for product data in product management. They distinguish the following higher-level categories: product planning data, production data, product control data, interface data, customer management data, customer interaction data, market environment data, feedback data, product operating data, and web engagement data. As their taxonomy covers also non-software products, not all data points are relevant for software products, however, the taxonomy gives an overview of the breadth of available data that product managers should effectively take into account when being data-driven. The complexity is further increased by each category having various data points.

In addition to product data, organisations have various other data categories that are used to support product-related decisions, such as business data, process data, project data, and operational data [18]. Business data is used in business decisions such as details about market trends, customer behaviour, and sales data. Process data gives an overview of the workload, bottlenecks, and team performance within the software development process. Project data relates to the overall software project including risks, budget, resources, and schedule. Finally, operational data is company-level data related to strategy and overall data management such as data privacy information.

3.2 Challenges Related to Software Product Data

This chapter discusses challenges that arise from the researched literature. These challenges are related to product data itself and the collection and use of it. The identified challenges are presented in Table 3.2.

Effective data management can lead to improved decision making, business performance, and understanding of customers [18]. While various data types exist, the usage is often non-systematically focused on only some data types [10], confirming already existing beliefs instead of disproving them or discovering new insights [11], or not used at all in decision making [15]. There have also been identified issues with the effective use of data through the various stages in the product development

Table 3.2: Challenges related to software product data.

Challenge	**Description**	**References**
Data infrastructure	Data is collected and stored in different places and in different formats making using and integrating data into tools difficult.	[15][17]
Data availability	Data is not collected or is only collected from limited sources leading to there not being (enough) data to analyse or use.	[9][10][15][18]
Amount of data	There is more data generated than can be analysed and used, causing issues with storing and effective use of data.	[11]
Data complexity	Data is gathered from various sources and in various formats making integrating different data types difficult.	[10][14][15][17][18]
Data quality	Data is formatted poorly, is incorrect or there is missing data.	[11][14][18]
Data accuracy	Data is not representative of the reality, producing incorrect or skewed results.	[10][18]
Data privacy	Privacy limitations restrict the collection and use of data leading to ineffective and incomplete data collection and usage.	[8][15][16]
Poor data tools	Usable tools that integrate various data types for effective analysis are not available.	[8][9][15][18]
Poor data skills	Skills needed to understand and analyse data and use data tools are not available.	[8][9][18]
Turning data into insights	Analysing and procuring actionable insights understandable by various stakeholders from raw data is difficult and burdensome.	[10][14][17][18]
Data visualisation	Visualising complex data into easily undestandable format for communicating with stakeholders is difficult.	[14][15]
Effective use of data	Use of data is focused on types that are easily available and/or well understood while most data is ignored.	[3][10][11][14][15][16]

[16]. Moreover, data is used more effectively for less complex problems and more complex problems also require a larger variety of data [3].

Springer & Miler [9] systematically mapped out software product management challenges in their study. While the most frequent challenges were related to value measurement, frequently changing priorities, and technical debt, various issues related to data were also identified. The data-related problems included lack of user data, unusable analytics setup, lack of market research, limited access to users for research, lack of research resources, lack of user research, and lack of analytics skills. Out of these issues lack of research resources (65.9%), lack of user research (61.4%), and lack of market research (55.7%) were reported as most frequent, while unusable analytics setup (44.3%) and lack of analytics skills (43.2%) were listed as least frequent. In terms of severity of the issues, lack of user research (64.8%), lack of research resources (63.6%), and limited access to users for research (59.1%) were most severe, while unusable analytics setup (44.3%) and lack of analytics skills (43.2%) were least severe. Thus, the study proclaims that the biggest issues for data-driven product management are related to research resources and user research, while all issues are more or less common and severe. However, the lack of user data was perceived as more frequent by experienced product managers, which could indicate that experienced software product managers are more inclined to use hard data.

Other sources echo similar results from software product management. According to Holmström Olsson & Bosch [11] the main challenges in data-driven product management include handling increasing amount of data, effective data usage, and ensuring data quality. Issues with data management and practices can hinder effective decision making via creating complexity, confusion, misinformation, and data privacy issues [8]. Data privacy and confidentiality is mentioned as an increasingly relevant challenge with evolving technologies [16] and stakeholder communication [15]. A case study by Holmström Olsson & Bosch [1] found out that data is not effectively used for development and improvement of functionality, but the focus is rather on quality assurance, performance monitoring, and diagnostics. Fichtler et al. [10] argue that differences in availability and accuracy of different data types can make comprehensive analysis difficult. Another limiting factor for the effective exploitation of data analytics is a lack of analytical skills and tools [8][18]. Existing tools are considered to be focused on technical data and targeted towards developers, ignoring the higher level product goals that would be more relevant for software product managers [15].

Similar data-related issues are reported in software development as in software product management: data complexity, data collection, data analysis, and storing data [17]. Research by Rique et al. [15] also identified issues with data complexity, availability and infrastructure, including integrating new data sources into systems, structuring data within systems, and ensuring effective data collection during development. Functionality of the data infrastructure also determines the effectiveness of continuous experimentation [3]. Data comes in many formats from numerous places and analysing such heterogeneous data in an efficient manner portrays difficulties for analysts and tools themselves [17]. An additional challenge is presented with data from dynamic and unintended sources, where data is available at large in nearly unlimited sources and formats [14].

Further, agile software development has been reported to have issues with data management, including collection and quality as well as procuring insights from raw data [18]. The survey by Fawzy et al. [18] highlighted the difficulty of integrating various data sources while maintaining accuracy and completeness of data. Automating data collection and analysing complex sets of data to achieve real-time analysis was also considered to be challenging. They further emphasize that issues with data management negatively impact all areas in an agile software project including gathering requirements, planning sprints, daily meetings, retrospectives, testing, and quality assurance.

Other relevant data issues include visualising data [14][15] and recognising irrelevant data from both quantitative and qualitative data [14]. Especially data expressed in human language is difficult to analyse for relevancy due to its ambiguity and context-dependence. Data visualisation increases actionability and ensures that data is used in decision making [15]. Visualising data is not an established practice in defining requirements despite its importance for identifying and communicating insights, patterns, and trends [14].

3.3 Emerging Data Practices

The following chapter introduces emerging data practices from the fields of software product management and software development. The recent developments in technology are especially relevant for the future of data-driven software product decision making. The emerging practices found in the literature are presented in Table 3.3.

Advancements in technology allow data-driven software product management to become more efficient and effective. These technologies provide opportunities to shorten the feedback loop and support managing rapid changes and uncertainty [11]. Especially machine learning (ML) has been highlighted as a recent development to improve data analytics for software products [15]. Holmström Olsson & Bosch [11] argue that companies are already employing a mix of requirements-driven, data-driven and artificial intelligence (AI)-driven approaches in selecting and prioritising product functionalities. Moving from traditional requirements-based techniques to data, ML- and AI-based practices increases returns on development investments.

Harkonen et al. [8] define ways in which Industry 5.0 (I5.0) affects data practices within product management. Industry 5.0 refers to a new era of technological capabilities including modern approaches such as human-computer collaboration,

Table 3.3: Emerging practices in managing software product data.

Emerging practice	**Description**	**References**
Machine Learning (ML)	ML models independently or with guidance crawl and analyse massive amounts of data.	[1][8][11][14][15][17]
Artificial Intelligence (AI)	AI models procure insights from vast data, are able to summarise and visualise complex data, and can quickly build functionality prototypes.	[1][8][11][16]
Natural Language Processing (NLP)	NLP models make analysis of data expressed in human language more efficient.	[14][17]
Analytics automation	By automating data analysis with technologies such as ML, AI and NLP procuring actionable insights from data becomes more efficient.	[8][14]
Internet of Things (IoT)	IoT data increases data versatility by complementing existing data sources.	[8][14]
Development Operations (DevOps)	DevOps streamlines software development processes shortening the feedback loop.	[1][11][16]
Digital ecosystems	Software product's dependence on other digital systems complicates data collection and analysis while providing digital business opportunities.	[1][11]

ML, AI, and internet of things (IoT). According to them, I5.0 remarks a movement towards focusing on sustainability, resilience, hyper personalisation, and data utilisation. These emerging changes require adapting product management and business processes to be more flexible and scalable. Utilising ML, AI and IoT technologies to automate and optimize data analytics are suggested to lead to better insights, decision making, and shared product understanding. They also play an important role in supporting data visualisation, which helps stakeholders act in a data-driven way beyond core product management. Finally, the I5.0 technologies open new opportunities for future anticipation, which is complicated and fairly uncertain with traditional data analytics methods.

Holmström Olsson & Bosch [1][11] likewise highlight the emerging role of AI and data but additionally raise development operations (DevOps) and digital ecosystems as important practices affecting software product management. DevOps aims to improve the efficiency of software development through process automation, continuous integration, and continuous deployment, which enable software product managers to gain feedback from the product quickly [16]. Digital ecosystems are built of various interdependent digital products and systems interacting with each other, building digital business environments [1]. These trends are argued to shift the focus of product management from predicting outcomes to hypotheses testing and experimenting [1][11]. However, current software product management frameworks are insufficient for supporting the introduction of the emerging technologies and practices. Thus, software product managers require support for moving towards using hypotheses as the basis for product development instead of traditional requirements.

AI is portrayed as a complementary tool for supporting humans during software development [1]. It can ensure efficient value creation through continuously monitoring and optimizing system functionality [11]. AI is additionally seen as an opportunity to significantly improve A/B-testing performance [16]. Holmström Olsson & Bosch [1] introduce the SPM4AI Framework for navigating human-AI-interaction in the field of software products. In situations of high certainty and stability the options for SPM function are to either to start building functionality based on human-defined requirements or model available data with ML or AI. With an evolving product landscape, the goal is to explore and develop alternative solutions. However, if there is available data, ML or AI models can be retrained with new data, which is a fast way of testing alternatives, and with the efficiency comes the opportunity of trying a larger number of alternatives than would be possible with traditional software development. In an environment of low certainty, humans need experimentation often to even know whether a functionality is necessary, to which low-effort experimentation methods such as A/B-testing provide a solution. AI models may be used to create higher fidelity demos for experimentation with the level of efficiency that would not be possible traditionally. In all cases of certainty, the human and AI practices are considered to be used in a complimentary and parallel manner. The AI practices are highly dependent on availability, amount, and accuracy of data, which highlights the importance of high quality data management.

SPM4AI framework was further elaborated with Strategic Digital Product Management (SDPM) framework [11] (Figure 3.2), where the approach to decide functionality is selected based on the certainty dimension and the approach dimension. Traditional approaches rely on requirements elicited from qualitative data in high certainty, on software updates in evolving certainty, and on mock-ups in low certainty. Data-driven approaches use requirements elicited from analytics in high certainty, anomaly detection within analytics in evolving certainty, and A/B-testing

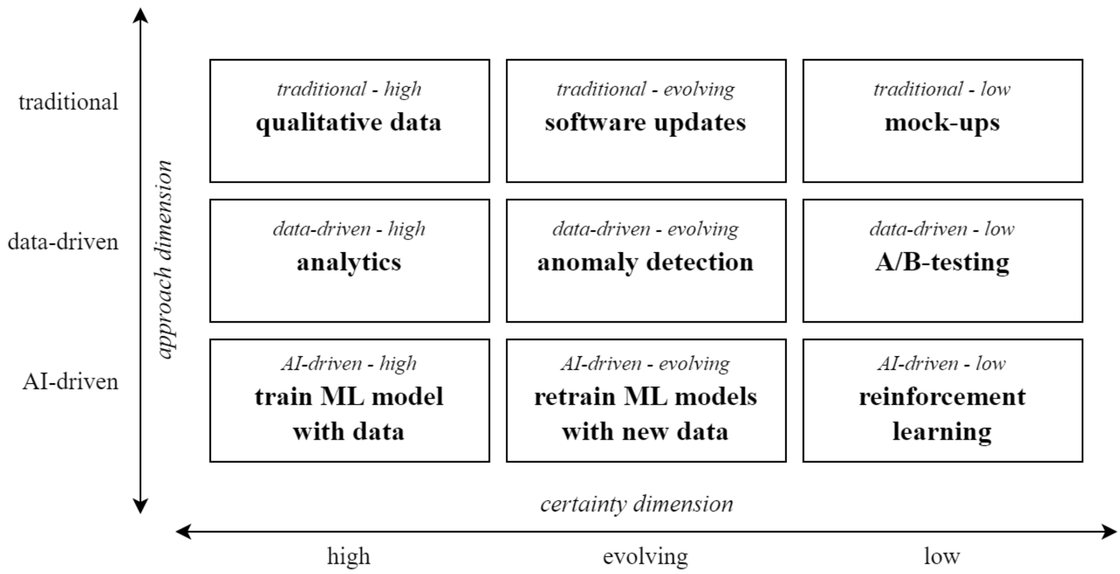

Figure 3.2: SDPM framework based on Holmström Olsson & Bosch [11].

in low certainty. Finally, AI-driven approaches benefit from using data to train an ML model in high certainty, retraining ML models with new data in evolving certainty, and reinforcement learning in low certainty. Similar to SPM4AI framework, the SDPM framework suggests flexibly choosing and changing between approaches based on each use case. However, moving from traditional approaches towards data- and AI-driven approaches requires generous amounts of high quality, accurate data.

In software development, Natural Language Processing (NLP) and ML models based on NLP have been proposed as ways of increasing the efficiency of requirements gathering [17]. While quantitative data can indicate issues, the needs and wishes of users are often best communicated with qualitative feedback and transforming sets of qualitative data into actionable requirements calls for sophisticated NLP methods [14]. Automating analytics with the help of ML, AI, and NLP could in the future significantly enhance data-driven software product management by increasing efficiency and effectiveness. Currently analytics automation is focused on supervised ML to elicit requirements from qualitative data, especially reviews and feedback. Overall, effective analytics automation can support in filtering irrelevant data and identifying themes, speeding up the analytics process. Automating analytics at a large scale could increase the importance of dynamic data, including IoT-generated data, as a source for requirements. Such data can bring up undiscovered insights and ensure timeliness of insights.

Evolving technologies could pose software product management to become increasingly challenging. Managing data-intensive AI systems is vastly different from traditional tools [11]. Evolving technologies and software delivery practices require rapid adaptation from the development organisation [1]. Springer & Miler [9] identified that technology-related software product management challenges were not mentioned a decade ago but are now some of the most prevalent issues for software product management. However, they also see technologies e.g. automated testing and continuous deployment as opportunities to solve the prevalence of technology-related challenges in the near future.

4 Discussion

This chapter discusses the findings of this literature review further and in light of previous research about product data and data practices in software product management and software development. The findings are reflected to practical implications for software product managers.

Currently there are few data practices in software product management and the data practices used in software development have been exclusive for development needs. Software development focuses largely on how software developers use and analyse data instead of considering the managerial side of software. The usage of product data for higher level, strategic decision making does not receive as much attention as it does in software product management, where there has been much emphasis on it. There exists a gap in using product data between the functions despite them becoming increasingly intertwined and included in the same frameworks. Based on this study the functions also share same data challenges and are both looking into the same solutions emerging from recent developments in technology. Therefore there is a need for shared understanding and practices for working with product data to achieve better decisions, outcomes, and higher value. There is also a need for a shift in attitudes towards product data and its usefulness to achieve effective adoption of data-driven software product management, since earlier research also suggests that software product managers seem to not be interested in technical data despite its usefulness [19].

Academia has already previously widely recognised the same challenges related to software product data as were presented in Table 3.2 [2][4][5][6][7][19]. For those data challenges in the table that had more than one reference, all were mentioned in both software product management and software development literature, except for data infastructure and data visualisation which were only mentioned in software development research. The research sample, however, is limited and thus it can be assumed that most if not all data challenges are prevalent in both fields. The most relevant issues relate to the effective use of product data (6 references) and complexity of data (5 references). These, as well as the other challenges, can be seen to also affect each other. For example data complexity makes using data difficult, which would require proficiency in data skills and fitting tools to handle the large amounts of data and elicit insights from them. Visualising data and procuring actionable insights are the key in ensuring that software product data are used in decision making by various stakeholders. Large amounts of complex data are more easily understood and internalised with visualising. Data visualisation is best achieved with dashboards which play an important role for software product managers in analysing data and communicating insights to stakeholders, as shown by previous research [19].

Lack of proper tools for analysing software products can be seen as one of the most impactful issues both intrinsically but also as a source for other data challenges. Having proper analytics tools is of major importance in order for software product managers to use product data effectively. Current data tools are not useful because they focus on quantitative data, which does not give enough deep insights. These tools do not tell what the software product manager should do, which is the most important consideration for software product managers. Thus, there is a disconnect between managerial needs for product data tools and what the existing tools provide: an effective tool for software product management helps managers answer questions of "how" and "why" instead of "what". An effective product analytics tool could handle massive amounts of data, integrate different data formats, notice and ignore incorrect data, ensure data privacy, reduce the need for advanced data skills, produce context-dependent insights, and use visualising to ensure data is effectively used. Such a tool could solve the issue of data tools being used reactively, where managers only dive deep into analytics when they have a feeling of lacking understanding or there is a specific problem to be solved.

Overall, there seems to be a fundamental problem with how data management is approached. Organisations have long been collecting data for the sake of collecting without a clear data strategy. Product data collection should stem from a need for solving a problem instead of arbitrary collection which could solve various data issues. There is a wide gap between data collection and the effective use of data. The haphazard data collection is becoming more and more limited due to data privacy issues. Data privacy is perhaps the most difficult data challenge to solve since it is not just dependent on the company policies but also governing legislation. For example GDPR has been stated to significantly restrict data collection in software projects [20]. Thus, an already difficult feat of analysing and using product data is becoming evermore complex to which emerging practices in product data management could provide solutions. ML/AI models could support with data management by anonymising data before it can be viewed by any stakeholder or it could procure insights from data without giving access to the raw data. Of course in such cases there arise questions of whether the ML/AI model is able to protect the data and that the data used by the models still must be stored somewhere.

Application of emerging practices identified in Table 3.3, especially machine learning, have been gaining traction already since the end of 2010s [7]. Similar to data challenges, the emerging practices were all mentioned in both software product management and software development literature, with the exceptions of NLP, which was only mentioned in software development research, and digital ecosystems, which was only mentioned in software product management research. Thus, it seems that both functions are looking into these new opportunities for enhancing product data practices, with machine learning at the forefront. ML/AI models provide solutions to overcoming issues with amount of data, data complexity, poor data skills, even data quality, all while being able to turn data into insights and visualise data. These tools show promise in solving data challenges with the ability to crawl massive amounts of complex data as well as automating coding, prototyping, and data generation. They can also assist software product managers with hypothesis testing for non-released features. ML/AI models could also be trained to anticipate the future and used to try out features before releasing with fast prototyping. With the ability to generate code and data, and automate testing, ML/AI models can significantly improve the efficiency of developing and managing software products. Thus, these practices could support the analysis of software products that are in low maturity stage. With the addition of NLP methods, the analysis of data expressed in human language can also be enhanced. Such qualitative data in the form of e.g. feedback and reviews have been considered to be of major importance in providing deep insights in managing software products. The overall process of automating analytics is considered to bring major benefits for solving data-specific issues as well as supporting decision making.

With the integration of software product management function into existing software development frameworks, there could be an introduction of a more profound start-to-end data approach, where all of product data, both quantitative and qualitative, pre- and post-deployment, are monitored together. The new technologies proposed by the emerging practices could be the solution to enable such a multidimensional data practice to be integrated into frameworks overarching both software product management and software development. Such effective framework could provide significant improvements for creating and managing software products, which could in turn increase the product's value and further business value for the organisations. There is also potential for improving engagement with the product from the sides of software product management and development as well as product users as all key stakeholders enjoy improved understanding and communication works better with an effective feedback loop.

Still, achieving benefits from product data is not possible only by solving the data challenges, providing tools that use the emerging technologies, and putting the responsibility of being data-driven to each software product manager. The promotion of data-driven software product management is also a responsibility of organisations and wider culture as a whole. Having a data analyst within the software product team has been shown to promote the use of data and help to avoid subjective decision making [19]. Organisations should ensure such a position is filled in software product teams to achieve benefits of data-driven software product management. Additionally it should be noted that going full on with data is neither the key to success. Despite the importance of using product data in making decisions about software products, academia has suggested being informed by data instead of driven by it, since complex contexts and domains require human interpretation [7]. Thus, a healthy balance between data analytics and other evaluation methods must prevail.

5 Conclusion

Software product management and software development are both functions involved in managing software products. This literature review explored the recent research related to software product data in both functions. Existing frameworks have mostly set these as separate functions, especially when it comes data practices and product data. Still in practice these functions overlap in many situations and both could benefit from borrowing data practices from the other function. So far software product management has had a higher level managerial focus on when it comes to managing and using product data. Software products generate copious amounts of complex data, but software product management lacks practices, frameworks, and tools to answer the product-related questions that are important for them: "why" and "how". On the other hand, software development has integrated data more into their frameworks, but the focus is on technical data that answers the question of "what". Aligning these functions, their existing practices and needs that have yet been fulfilled could provide the people working closely with the product better understanding of the product on a micro and macro levels.

Effective use of product data and successful procurement of actionable insights provide benefits to external stakeholders as well. As managing and using product data is often difficult for the professional involved with the product on a day-to-day basis, it is even more complex for stakeholders outside the core product team. Thus, the benefits gained from effective data management and use extend outside the product team and to external stakeholders. Currently there are various challenges related to the management and use of product data, which must be overcome to achieve better product understanding and decision making. Developments in technology and practices are showing promise to solve many of the problems, with machine learning at the forefront. Emerging practices can massively increase the efficiency of product data management and use on a level that could not be achieved with human skills.

The prevalent data challenges, emerging practices portraying various solution possibilities, and the existing data-driven processes shed light for the research questions presented in Chapter 1. As for the question on what is the role of product data in SPM (RQ1), it seems that product data is collected extensively and used with face value, but further analytics, integration, and insights are lacking. Currently the SPM function discusses using product data on a high level, referring to data being the key to value creation while simultaneously missing shared practices or frameworks for effectively using product data. Various challenges exist related to product data, of which data availability, poor data tools, poor data skills, and effective use of data were most relevant for software product management. In SPM research the focus of emerging practices to solve these issues has been on various ML and AI models.

Software development literature echos similar findings for the question of what are the relevant themes related to product data in SD (RQ2). However, as opposed to the lack of using product data in SPM, software developers seem to be active in using product data and the use of data in software development has been researched a lot. Nevertheless, software development function suffers the same challenges with product data as SPM, with the focus being on data complexity and effective use of data. Out of the emerging practices ML has received most attention with NLP as a second.

Since the data challenges and emerging practices related to product data seem to be mostly shared with software product management and software development functions, there is little gap for finding synergy improvements from one side to the other. Thus, as for the question of how can SPM and SD product data practices be better aligned (RQ3), it seems that in many ways they are already aligned in the academic field. The biggest differences between the functions come from the background and point of view. Software development has been an established function for decades and the use of product data has been widely researched. The approach to product data seems to also be on focusing on the technology, identifying bugs, and getting information about the "what". On the other hand, software product management as a discipline is quite new and research on the data practices within the function are few. Thus, the use of product data is considered more as an opportunity and a concept. The approach towards product data is also focused more on understanding users on a larger scale, getting insights into the goals and needs of the users, instead of being interested in the product itself. Software product managers are therefore looking for answers to the questions of "how" and "why".

The future of making the management and development of software products more data-driven shows promising. However, data is not a fix-all solution, but professional opinions, expertise, and domain knowledge are still required to support with the analysis and decision making. Further empirical research is required to identify ways in which software product management and software development functions could be better aligned in terms of product data. Especially emerging practices must be developed and tested in practice to confirm the suggested benefits those bring along. This research has also been limited to one framework and thus there is a need to survey current frameworks from both functions to create a broad overview of how the functions are seen to establish themselves together and separately in terms of product data.

References

[1] H. Holmström Olsson and J. Bosch, "Strategic digital product management in the age of ai", *Proceedings of the 23rd International Conference on Software Business, ICSOB 2023*, pp. 344–359, 2024. DOI: 10.1007/978-3-031-53227-6_24.

[2] N. B. Moe, M. Berntzen, A. Barbala, and V. Stray, "Software product management in large-scale agile", *Agile Processes in Software Engineering and Extreme Programming (XP 2024)*, pp. 53–69, 2024. DOI: 10.1007/978-3-031-61154-4_4.

[3] R. Ros, E. Bjarnason, and P. Runeson, "A theory of factors affecting continuous experimentation (face)", *Empirical Software Engineering*, vol. 29, 1 2024. DOI: 10.1007/s10664-023-10358-z.

[4] R. P. L. Buse and T. Zimmermann, "Information needs for software development analytics", *2012 34th International Conference on Software Engineering (ICSE)*, pp. 987–996, 2012. DOI: 10.1109/ICSE.2012.6227122.

[5] F. Fotrousi, K. Izadyan, and S. A. Fricker, "Analytics for product planning: In-depth interview study with saas product managers", *2013 IEEE Sixth International Conference on Cloud Computing*, pp. 871–879, 2013. DOI: 10.1109/CLOUD.2013.33.

[6] F. Fotrousi and S. A. Fricker, "Software analytics for planning product evolution", *Software Business (ICSOB 2016)*, pp. 16–31, 2016. DOI: 10.1007/978-3-319-40515-5_2.

[7] C. Matthies and G. Hesse, "Towards using data to inform decisions in agile software development: Views of available data", *Proceedings of the 14th International Conference on Software Technologies (ICSOFT 2019)*, pp. 552–559, 2019. DOI: 10.5220/0007967905520559.

[8] J. Harkonen, J. Koskinen, and T. Kotilainen, "Industry 5.0: Data analytics & product management perspective", *2024 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2024. DOI: 10.1109/IEEM62345.2024.10856990.

[9] O. Springer and J. Miler, "A comprehensive overview of software product management challenges", *Empirical Software Engineering*, vol. 27, 5 2022. DOI: 10.1007/s10664-022-10134-5.

[10] T. Fichtler, L. Kirchberg, K. Grigoryan, C. Koldewey, and R. Dumitrescu, "Antecedents of data-driven product management – data taxonomy and cluster & software categories", *2024 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, 2024. DOI: 10.1109/ICTMOD63116.2024.10878154.

[11] H. Holmström Olsson and J. Bosch, "Strategic digital product management: Nine approaches", *Information and Software Technology*, vol. 177, 2025. DOI: 10.1016/j.infsof.2024.107594.

[12] A. Maglyas, U. Nikula, A. Smolander, and S. A. Fricker, "Core software product management activities", *Journal of Advances in Management Research*, vol. 14, pp. 23–45, 1 2017. DOI: 10.1108/JAMR-03-2016-0022.

[13] I. S. P. M. Association, *Knowledge - ispma*, 2025. [Online]. Available: https://ispma.org/bok/.

[14] S. Lim, A. Henriksson, and J. Zdravkovic, "Data-driven requirements elicitation: A systematic literature review", *SN Computer Science*, vol. 2, 1 2021. DOI: 10.1007/s42979-020-00416-4.

[15] T. Rique, M. Perkusich, E. Dantas, D. Albuquerque, K. Gorgônio, and H. Almeida, "On adopting software analytics for managerial decision-making: A practitioner's perspective", *IEEE Access*, vol. 11, 2023. DOI: 10.1109/ACCESS.2023.3294823.

[16] F. Quin, D. Weyns, M. Galster, and C. Silva, "A/b testing: A systematic literature review", *Journal of Systems and Software*, vol. 211, 2024. DOI: 10.1016/j.jss.2024.112011.

[17] A. Henriksson and J. Zdravkovic, "Holistic data-driven requirements elicitation in the big data era", *Software and Systems Modeling*, vol. 21, pp. 1389–1410, 4 2022. DOI: 10.1007/s10270-021-00926-6.

[18] A. Fawzy, A. Tahir, M. Galster, and P. Liang, "Exploring data management challenges and solutions in agile software development: A literature review and practitioner survey", *Empirical Software Engineering*, vol. 30, 3 2025. DOI: 10.1007/s10664-025-10630-4.

[19] J. Paasila, "Data-driven software product management: What is the role of a modern software product manager and how data-driven are they in making decisions?", M.S. thesis, Aalto University School of Business, 2024. [Online]. Available: https://urn.fi/URN:NBN:fi:aalto-202408255612.

[20] A. Barbala, T. Sporsem, and V. Stray, "Data-driven development in public sector: How agile product teams maneuver data privacy regulations", *Ag-*

ile Processes in Software Engineering and Extreme Programming (XP 2023), pp. 165–180, 2023. DOI: 10.1007/978-3-031-33976-9_11.