Strategic Digital Product Management in the Age of AI

(article starts on next page)

Strategic Digital Product Management in the Age of AI

Helena Holmström Olsson[1(✉)] and Jan Bosch[2]

[1] Malmö University, Malmö, Sweden
`helena.holmstrom.olsson@mau.se`
[2] Chalmers University of Technology, Göteborg, Sweden
`jan.bosch@chalmers.se`

Abstract. The role of software product management is key for building, implementing and managing software products. However, although there is prominent research on software product management (SPM) there are few studies that explore how this role is rapidly changing due to digitalization and digital transformation of the software-intensive industry. In this paper, we study how key trends such as DevOps, data and artificial intelligence (AI), and the emergence of digital ecosystems are rapidly changing current SPM practices. Whereas earlier, product management was concerned with predicting the outcome of development efforts and prioritizing requirements based on these predictions, digital technologies require a shift towards experimental ways-of-working and hypotheses to be tested. To support this change, and to provide guidelines for future SPM practices, we first identify the key challenges that software-intensive embedded systems companies experience with regards to current SPM practices. Second, we present an empirically derived framework for strategic digital product management (SPM4AI) in which we outline what we believe are key practices for SPM in the age of AI.

Keywords: Strategic digital product management · DevOps · Data · Artificial intelligence · Digital ecosystems · Digitalization · Digital transformation

1 Introduction

The role of product management is critical for the success of any product. As recognized in [7], the product manager holds responsible for product requirements, release definition, product release lifecycles, creating an effective product introduction team and preparing and implementing the business case. Similarly, [27] describes software product management (SPM) as a crucial discipline that encompasses the activities and responsibilities involved in creating, delivering, and maintaining software products. In addition, and as pointed out in [7], the product manager owns the business case and assures that a product release delivers the expected value to customers as well as to the business. In practice,

and especially in the software-intensive embedded systems industry, SAFe is one of the most common frameworks for product strategy, planning and roadmapping[1]. During recent years, it has become widely adopted by companies that wish to scale their agile practices, accelerate value delivery and shorten feedback loops to customers. Although research on the benefits of adopting SAFe is still scarce, it remains the predominant framework for software organizations that seek to accelerate value delivery to customers. In addition to SAFe, there are several frameworks and models for supporting and improving software product management practices. As a few examples, the ISPMA framework provides a holistic view on the activities of software product management[2], the SPM reference framework identifies key process areas as well as the stakeholders and their relations [35], the SPM competence model outlines key capabilities a software organization should implement to improve SPM maturity [2], the market-driven product management and requirements engineering model (MDREPM) enables software process improvement and process assurance [13] and the 4CC model provides a blueprint for re-engineering product development management practices [30]. Also, there are numerous papers outlining key success factors for software product management [8] and SPM best practices, e.g., [10,33,36].

However, although there is prominent research on software product management and the importance of this discipline, there are few studies that explore how the role of product management is rapidly changing due to recent, and profound, trends that come with digitalization and digital transformation. As concluded in our previous research [5], digital technologies change development organizations and how these operate. In our view, digital transformation has significant implications on the software product management. Similarly, [21] recognize how the principles of how software products are introduced and delivered to customers are changing rapidly. Although software product management can, and in our view should, be considered part of the field of software engineering, in this paper we use these terms as separate. In the remainder of the paper, we use the term software product management to refer to decisions concerning what to build and why it should be built. We use the term software engineering to refer to decisions and activities concerning how to build the prioritized functionality.

In this paper, we explore how key trends such as DevOps, data and artificial intelligence (AI), and the emergence of digital ecosystems challenge and fundamentally change current SPM practices. Our research builds on multi-case study research in companies in the embedded systems domain that experience rapid changes in the business environments in which they operate and as a consequence, need guidelines for how to approach and reason about their SPM practices going forward.

The contribution of this paper is two-fold. First, we identify the key challenges that companies in the software-intensive embedded systems domain experience with regards to their current SPM practices. Second, we present an empirically derived framework for strategic digital product management (SPM4AI) in which

---

[1] https://scaledagileframework.com/.
[2] ispma.org.

we outline what we believe are key practices for SPM in the age of artificial intelligence (AI).

The remainder of this paper is structured as follows. In Sect. 2, we review literature on software product management and framewroks that are currently used to support this role. Also, we outline key trends that we see challenge current SPM practices. In Sect. 3, we provide an overview of the research approach we used and the case companies involved in our study. In Sect. 4, we present the empirical findings. In Sect. 5, we present the 'Strategic digital Product Management' framework (SPM4AI) in which we outline what we believe are key practices for SPM in the age of AI. In Sect. 6, we discuss threats to validity. In Sect. 7, we conclude the paper.

2 Background and Related Work

2.1 Software Product Management (SPM)

Engineering is concerned with building systems and with activities such as e.g., requirements engineering, designing an architecture, developing software, implementation of software, testing and validation of the system and finally, release to customers. However, whereas engineering is concerned with 'how' to build systems, there is another activity concerned with 'what' to build and even more important, 'why' we should build the system in the first place. This activity is typically referred to as product management and in the context of software-intensive systems as software product management. Over the years, numerous studies have explored the activities involved in software product management and the role of the software product manager. In [8], the authors conclude that the SPM role is critical and that with a consistent and empowered product management role, the success rate of projects in terms of schedule, predictability, quality and project duration improves. In [2], a product manager is referred to as the *"mini-CEO of an organization"* as they are positioned at the center of the organization where they keep in contact with all stakeholders to ensure that they work towards the same goal. In [28], the author discusses how proper product management processes improve resource management efficiency, lead to increased business growth, better budget control, higher user satisfaction, increased release predictability and faster release cycles. As depicted in [12], software product management is the role responsible for what the product is, how it works, whom it serves and how it affects the company and its customers. As a comprehensive summary, [32] outline key product management practices in a framework involving management processes, support processes and software lifecycle processes. As can be seen in the studies mentioned above, and if looking at the impressive body of knowledge in the field, the importance of this role is only increasing.

2.2 SPM Frameworks

There are several frameworks and models that provide support for software product management. With a focus on how to effectively scale agile practices,

SAFe has become one of the most common and widely adopted frameworks in industry [https://scaledagileframework.com/]. In the most recent version, product management is described as the function responsible for defining desirable, viable, feasible, and sustainable solutions that meet customer needs and as the function supporting development across the product life cycle. In [29], the authors conclude that increased transparency, alignment, quality, time to market, predictability and productivity are the perceived benefits of SAFe, while the challenges are associated with resistance to change and controversies with the framework.

In addition to SAFe, there are prominent frameworks such as e.g., the ISPMA framework [ispma.org]. This framework provides a holistic view on the activities of software product management with the intent to establish and improve SPM practices in organizations. In [15], the authors build on the ISPMA framework when providing best practices for product strategy, product planning, strategic management and orchestration of the functional units of the company. In [11], the framework is referred to as unique in that it integrates several key characteristics from previous frameworks for product management, as well as for student education purposes.

The SPM reference framework identifies key process areas as well as the stakeholders and their relations [35]. The framework is based on a review of state-of-the-art literature on software product management as well as experience from industrial case studies. In addition to this framework, the SPM competence model outlines key capabilities a software organization should implement to improve SPM maturity [2]. The model provides an overview of four business functions that are important to SPM, i.e., portfolio management, product planning, release planning and requirements management, and the focus areas for each of these functions. Also, the model indicates the interactions that take place between different stakeholders and how information flows between roles and functions.

As yet another model, the market-driven product management and requirements engineering model (MDREPM) enables software process improvement and process assurance in market-driven software engineering [13]. The model targets the unique challenges that product development organizations operating in market-driven environments are facing and can be seen as both a best-practice guide and a process assessment framework.

Finally, the 4CC (Four Cycles of Control) framework combines business management and software product development, and takes both a long-term and short-term view to software product release management [30]. The framework involves the type, timing, and content of different product releases, and aims at providing a common understanding for how to organize software product development.

2.3 Key Trends that Challenge Current SPM Practices

Based on recent research, as well as our experience of working closely with companies in the embedded systems domain for more than a decade, we identify

three trends that have an impact on current ways-of-working and that challenge current SPM practices. Below, we detail these trends and the effect they have on SPM.

DevOps: The emergence of agile practices was key as the sprint model fundamentally changed the ways in which software was developed and delivered. These practices are now scaling and with the emergence of DevOps the entire feedback cycle with customers is shortened when bringing development and operations together [18]. For DevOps to be effectively adopted, technical transformations include, e.g., automated deployments using build and continuous integration tools, treating infrastructure as code, and continuous monitoring of infrastructure and system behavior in production. On the organizational side, it is crucial to build and strengthen a collaborative culture to successfully establish a straightforward communication and shared responsibilities [9]. With DevOps, also the role of the product manager changes. First, it becomes much more integrated with the engineering team as the ways-of-working shift from being specification-centric to more experiment-centric. Second, with DevOps systems are grown instead of built. Rather than defining the requirements and building the system to meet the specification, the focus shifts to defining outcomes and iteratively deploying functionality that support these. Third, with an experiment-centric approach, product managers can continuously measure the impact of development efforts and hence, adopt a more customer-centric approach to product development.

Data and AI: Digital technologies are transforming industry to an extent that we have only seen the beginnings of. Across domains, companies experience rapid changes to their existing practices due to the many opportunities these technologies bring. As recognized in e.g., [5,25], data and AI allow for continuous improvement of system functionality and hence, continuous value delivery to customers. In addition, and as recognized in [26], data and AI provide the basis for new digital offerings and recurring revenue streams. Finally, data and AI enable companies to shift towards customer KPI-based business models and two-sided markets [1,31]. With data and AI, the role of the product manager shifts from being concerned with predicting the outcome of development efforts and prioritizing requirements based on these predictions, towards adopting experimental ways-of-working, defining hypotheses to be tested and using data from products in the field for continuous monitoring and improvement of customer value.

Digital Ecosystems: As a recent trend, business environments are being recognized as digital ecosystems [16]. The concept of digital ecosystems is proposed as a new way to perceive the increasingly complex and interdependent systems that are being created and that are characterized by self-organization, scalability, sustainability and with business models in which the main revenue stream no longer consists of the production of a product that is sold to customers, but rather, provision of a combination of services and products to their customers [16,17]. From a product management perspective, digital ecosystems reshape the business ecosystems in which companies operate. With new innovation platforms

and digital marketplaces, software product development is rapidly shifting from focusing on internal scale, efficiency, quality and serving customers in a one-to-one relationship, to contributing to an ecosystem of multiple players [4].

3   Research Method

3.1   Case Study Research

Case study research has become an appreciated method in software engineering research as it allows for empirical investigation of contemporary phenomena. In [3], case studies are defined as information gathering from a few selected entities with little or no experimental control. Similarly, [34] emphasizes how case studies are useful when studying organizational contexts with complex and intertwined conceptual structures. In our study, we adopted a multi-case study approach to explore how key trends such as DevOps, data and artificial intelligence (AI), and digital ecosystems challenge current SPM practices. The findings we present are based on close collaboration with a selected set of companies in the embedded systems domain. All the case companies are members of a larger research collaboration in which industry and academia work closely together to help accelerate digitalization (www.software-center.se) . In what follows, we report on research in which we use company workshops and frequent check-in meetings conducted between January 2023 and September 2023 as the basis for our findings. It should be noted however, that we have been working with the case companies as part of the larger research initiative for more than a decade. This gives us the opportunity to use previous insights and experiences as valuable and complementary input also in this study.

3.2   Case Companies

The following case companies were involved in our study:

- **Case company A** is a networking and telecommunications company. For the purpose of this paper, we engaged with roles involved in software product management, engineering management and data analytics. We studied two different use cases. *Use case 1* is concerned with how to balance requests from a large and diverse customer group. *Use case 2* is concerned with how to effectively use data for continuous improvement of software products.
- **Case company B** is a company manufacturing vehicles. For the purpose of this paper, we engaged with roles involved in software product management, technology management and strategy lead. We studied one use case concerned with how to adopt A/B testing practices in large-scale systems development.
- **Case company C** is a food packaging and processing company. For the purpose of this paper, we engaged with roles responsible for data management and connectivity and software and systems engineering. We studied one use case concerned with using deep learning (DL) for managing system evolution.

– **Case company D** is a company manufacturing trucks. For the purpose
of this paper, we engaged with roles responsible for product management,
technology management and autonomous drive. We studied one use case con-
cerned with using reinforcement learning to improve system behaviors.

3.3   Data Collection and Data Sources

As the primary data source for this study, we engaged in workshop sessions at
all case companies. The workshop sessions lasted for 1–3 hours, involved 4–10
people and focused on current SPM practices, challenges imposed with digi-
tal technologies and best practices and strategies for how to address and mit-
igate these challenges. In addition to the workshops, we had bi-weekly and/or
monthly check-in meetings to review status of the initiatives and we contin-
uously discussed solution development and next steps. Our findings build on
company workshops and frequent check-in meetings conducted between January
2023 and September 2023. We have worked with several of the case companies
for more than a decade, and have reported on specific teams, products and chal-
lenges in previous work. However, in this paper the focus is on software product
management whereas in earlier publications we focused on software engineering
challenges. In total, we met with the case companies in 12 workshops (7 work-
shops in company A, 3 workshops in company C and 2 workshops in company
D). With company B, we interacted primarily by using frequent check-in meet-
ings (on-line) and e-mail conversations. The longitudinal nature of our research
allows us to capture not only our most recent experiences in the companies, but
also challenges and solutions that we have seen emerge over time as a result of
their long-term and on-going digital transformation. As part of the collaboration
with the case companies, we were able to follow several improvement initiatives
as well as internal discussions on how to rethink and reinvent the SPM role.

4   Findings

The challenges experienced in the case companies are due to the rapid pace
of digital transformation and the new technologies and ways-of-working that
come with digitalization. From the perspective of SPM, this implies that existing
frameworks are insufficient as these often fail in effectively supporting short
DevOps cycles involving continuous development and delivery of data and AI-
intensive system components. Below, we describe a selected set of use cases. Each
use case illustrates a key challenge that the case company experience and how
the company responded to this challenge.

4.1   Everything Starts with a Requirement

Challenge: The case companies develop systems that are safety-critical and sub-
ject to strict regulations and legislation. Due to this, the primary approach to

development in all case companies is a requirement driven approach. As the start of development, product management is responsible for collecting and specifying requirements as input for the software development teams. Over the years, and increasingly so with practices such as continuous deployment and data-driven development being introduced, a number of limitations have been recognized in relation to the requirement driven development approach. The assumption that customer requirements can be identified before development starts is the most questioned one and with an increasing amount of product and customer data available the traditional approach to requirements is rapidly changing.

Response: In the case companies, we notice that a requirement driven approach to development is well suited for situations in which features and functionality are well-understood, where there is a long-term agreement between the customer and the development organization and where there is less frequent change imposed on the system. However, when applied also in a fast changing environment the requirement driven approach falls short. This was confirmed in all case companies involved in our study and people report on use cases in which SPMs "create a false illusion of certainty" by taking a requirement driven approach also in situations characterized by uncertainty. Our research shows that a key challenge is to find alternative approaches and frameworks that support software product managers also in evolving and uncertain system contexts [6].

4.2   Balancing Exploration and Exploitation

Challenge: Case company A delivers systems to a large number of customers with very different needs. The role of product management is to inventory these needs, to combine, merge, and prioritize among them, and to present a roadmap with a set of requirements for the next release of the system. In this process, effective management of customer feedback is critical. However, and as reported in our previous work [23], the development of systems that serve a large customer group can easily lead to a tension between two conflicting interests. On one hand, the development organization seeks to achieve scale in terms of implementing as many new features to as many customers as possible. On the other hand, the development organization needs to show responsiveness to strategic customers. This requires the ability to balance exploration and exploitation which is a challenge in the companies we studied. In [23], we reported on the software engineering aspects of this by outlining the development organization and the structure of the software teams.

Response: From a SPM perspective, use case 1 in company A illustrates the challenge of balancing individual customer requests while at the same time serving a large customer. During the workshops in company A, we learnt that the most rewarding approach is to have some of the organization's development teams dedicated to specific customers that the product manager identifies as the most strategic ones. Based on the requests from these customers, teams explore new features, collect customer feedback and improve these features in an iterative

and incremental fashion. Once exploration of features is done with strategic customers, these features are adapted to generic customer needs and included in the planned releases. For the software product managers we talked to during this study, this approach allows for exploratory development of new features and the ability to respond rapidly to strategic customers, while over time having the benefit of exploiting these development efforts with the larger customer base. From the perspective of the software product managers involved in our study, the opportunities for exploration are rapidly increasing with large amounts of data, as well as AI technologies, being available.

4.3   Towards Testing of Hypotheses

Challenge: To manage situations with low certainty is a challenge that all case companies experience. Over the years, we studied cases where product management prioritized features that, in the end, where never used by customers or used so seldom that the development efforts could not be justified. To address this challenge, companies need support for experimental ways-of-working where teams use hypotheses instead of requirements as the basis for development as highlighted in data-driven development approaches. Although there is detailed advice for how to conduct A/B testing in online contexts, support for how and when to adopt these practices in large-scale embedded software development is scarce. Still, there are some examples from the companies we studied where experiments are run to support smaller improvements of features and where collection and analysis of customer and product data informs development.

Response: In company B, A/B testing is used on test vehicles with the intent to test two different versions of an energy optimisation software with customers. The test fleet consists of 28 vehicles and the company uses an experiment group design method, i.e., 'Balance Weight Matched Design', to address the challenge of having a limited sample size and increase the experiment power with small samples. In [19, 20], we present the software engineering aspects of these experiments and show that balanced groups can be produced even when the sample sizes are small. Our recent interactions with product managers in case company B confirm that experimentation is well suited for situations where there is a need to test different hypotheses and where the solution to a problem is unclear. Also, the company has started applying experimentation in innovation efforts as there is the need to test and trial with customers in order to identify the potential value of new digital services and offerings.

4.4   Maximizing Use of Big Data Sets

Challenge: The case companies collect massive amounts of data from their products in the field. This data is primarily used for diagnostics and quality assurance as well as for monitoring and improving product performance. Most companies experience a situation in which the amounts of data are growing exponentially

due to an increasing number of connected devices, an increasing number of sensors in these and an overall need to collect new types of data. In our experience, a common challenge is how to make effective use of data to support development and improvement of software functionality. In this area, existing frameworks are few and for most software product managers the opportunities data provides are also associated with several challenges.

Response: In use case 2 in company A, we studied how machine learning (ML) is used to improve paging. The paging feature is an existing feature in the audio stream that detects when the connection is poor. However, due to the increasing complexity associated with large telecom networks, and competing factors such as latency, resource consumption and number of paging requests, the intention was to explore to what extent the paging feature could be improved by using ML. From a SPM perspective, the use case illustrates the opportunity to have AI technologies complement and even replace human efforts during software development. Also, it shows how ML models can help realize system functionality and perform classification and prediction activities that would be challenging for humans to accomplish.

4.5 Managing Problem Domain Evolution

Challenge: The case companies operate in safety-critical environments where system quality and performance is key. Significant effort goes into continuous monitoring of system to ensure and improve their performance. While it could be argued that quality is important for any system, the systems we studied operate in contexts where failure could lead to severe accidents and even deaths. Therefore, ways in which quality can be assured and continuously improved are critical. At the same time however, internal resources are limited and all companies face challenges with regards to how to increase quality while maintaining, or ideally decreasing, costs involved in this.

Response: In case company C, we studied a use case where the company uses deep learning (DL) models to detect defects in packaging at each client site during processing. The architecture of this use case was presented in [14] where we show how a global model in the cloud is trained with the knowledge gained from local model training at each client site. The learnings from the cloud are fed back and shared to the client sites for inference using transfer learning. The data set consists of packages with different patterns, types and colours and with the DL approach the case company could optimize performance and minimize risks involved in the production line. From a SPM perspective, this allows for an effective way to enhance quality assurance of products while at the same time reduce efforts and costs involved.

4.6 Let the System Figure It Out

Challenge: With the rapidly growing interest in AI, the case companies we studied are looking for approaches that help them use these technologies to explore,

learn and adjust to changes in the environment in which their systems operate. This is of particular interest in contexts characterized by low uncertainty. As a common method, federated learning helps enable large-scale training of models on the device where the data is generated, but with the sensitive data remaining within the data's owner. The approach is generally applicable when the data is evenly distributed across devices. However, in the case companies involved in our study, data is typically not uniform. Also, the data subjects may have different characteristics from one another. The quality of the trained model may then be problematic.

Response: In company D, we studied a use case in which a team used reinforcement learning to explore the reward of introducing a new feature into existing autonomous trucks. In particular, the use case is concerned with monocular depth estimation and in a recent paper we present the software engineering aspects of this case by detailing the ML algorithm, the data sets and the simulations that were used [37]. From a SPM perspective, the reinforcement learning approach allowed for effective exploration of an action space to determine if there was sufficient reward to be accomplished by introducing the monocular depth estimation feature to existing autonomous trucks.

5   SPM4AI: Strategic Digital Product Management in the Age of AI

Software product management is concerned with determining what to build. The goal of this decision process is to maximize the return on the investment of the R&D resources. To accomplish this, the product manager is required to predict what the impact of a function or features on the customer, market and other stakeholders will be. However, predicting the impact of new functionality is far from trivial and traditionally the software product manager simply had to prioritize the content of a release based on their best understanding and assessment. With the emergence of DevOps, we get a new mechanism available: as the release frequency is so high, we can afford to experiment with new functionality before completing it. In this way, DevOps allows for building a slice of new functionality, get it out to some of the customers and use experiments to incrementally add and improve a feature. Experimentation is particularly important in cases where the certainty that a feature will add value is low. Research shows that potentially more than half of all features in a system are never used or used so seldom that the R&D investment was wasted [24]. Experimentation is a powerful approach to address this challenge as we can answer the question on whether functionality adds value with a much lower investment. A second dimension of decision-making is how to realize functionality. Traditionally, all functionality was realized using algorithmic code developed by software engineers. With the emergence of AI, it becomes increasingly feasible to train ML/DL models with available data. These models can then perform classification, prediction as well as other forms of inference.

The challenge of uncertainty and change over time also exists for ML/DL models. In some cases, the input data and the domain in which the system operates is rather static and it is sufficient to train a model once and deploy it. In many situations, however, the context in which the system operates evolves over time. In the context of ML/DL models, there are two basic approaches to accomplish evolution of models. First, one can monitor the performance of an otherwise static ML/DL model. When the performance of the model starts to decrease, this can be used to trigger retraining of the model. This is an effective approach to evolve ML/DL models in changing contexts with an element of human supervision. Although the trigger for retraining may be automated, in most cases there is a human who decides whether a new model goes live or not.

An alternative to retraining models is to use reinforcement learning. In this case, the algorithm is given a state space and an action space. Based on the action the reinforcement learning algorithm takes it receives a reward. Based on this, the algorithm learns, over time, what action is preferred in each situation. In an evolving system, the algorithm continuously spends a small amount of its time exploring. Consequently, when an alternative action is becomes more suitable over time, meaning the reward goes up, the algorithm will learn this and adjust its behaviour.
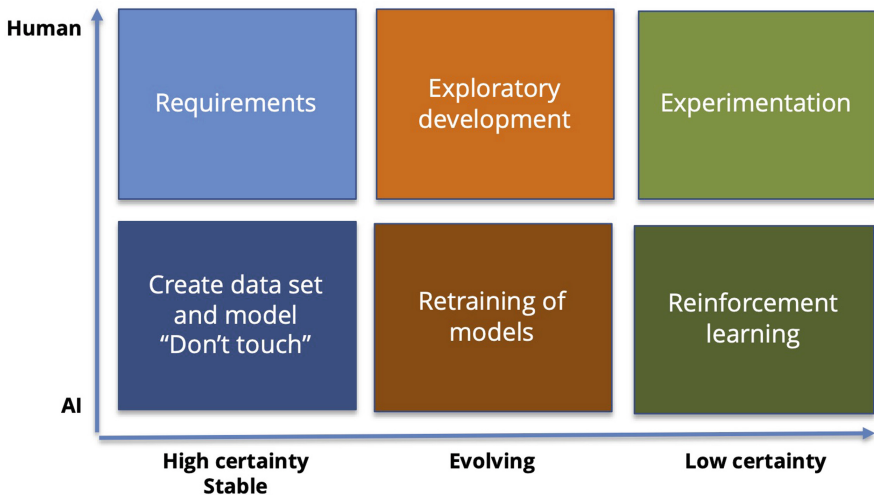
Fig. 1. SPM4AI Framework: six approaches

In Fig. 1, the insights that we developed during our study are summarized. When the functionality prioritized by the software product manager is considered to be stable and we have a high degree of certainty, we can either ask the R&D team to build the functionality based on the requirements or train a ML/DL model if there is data available.

In cases when the context in which the system operates evolves, the system has to respond to these changes. When the functionality is developed by humans, software product managers need to provide updated requirements for the development teams. The challenge is that even if it is obvious that the system needs to respond to changes, it may not be obvious how it should do so. To address this, we propose exploratory development where teams try alternative solutions to figure out the most rewarding path forward.

If the functionality is realized by ML/DL models, the typical approach is to retrain the model using the most recent data. There are challenges around when to retrain, define trigger points and how to ensure that appropriate monitoring is in place. Still, the opportunity to use ML models for managing system evolution is critical for SPM practices going forward as it comes with benefits that are hard to accomplish in traditional software development. If the degree of uncertainty is high to the point that it is not even clear that the functionality should be part of the system, companies need experimental approaches. As we shared earlier in the paper, many features in contemporary systems are never or hardly ever used. The goal of experimentation is to determine whether a new features should be part of the system at all. If the software product manager decides that a new feature or function should be realized through algorithmic code developed by a team, the suitable approach is to ask the team to conduct A/B experiments. The goal of the A/B experiments is to determine if there is sufficient value for customers or the company providing the system to its customers. In the case the software product manager decides that using an ML/DL model is the best way to realize the feature, reinforcement learning can be an effective approach to determine if there is sufficient reward to be accomplished.

To summarize this section, the role of product manager is to decide what to build in high degrees of uncertainty and a continuously evolving contexts. The framework we present identifies six approaches of realizing functionality that meets the specific constraints for each of the identified situations. In the end, the product manager needs to decide between these approaches based on his or her best understanding of the situation. In general our guidance is to select ML/DL models over algorithm-based development when feasible and to treat new functionality with more uncertainty then what one might believe. Both these guidelines allow for data driven decision making and reduced development efforts.

6   Threats to Validity

As the foundation for our understanding of the impact of digitalization on software product management practices, we reviewed contemporary research on this topic. Based on this understanding, we conducted multi-case study research in collaboration with companies in the embedded systems domain. As our primary data source, we collected data from workshops with key stakeholders within each of the case companies. To address construct validity [22], we shared our understanding of digital transformation, and the impact this has on SPM with all stakeholders involved in our research. With regards to external validity, we view

our research contributions as related to the "drawing of specific implications" and as a contribution of "rich insights" [34]. However, with the opportunity to study companies covering different industry domains we believe that the findings have the potential to be relevant also in other embedded systems companies with similar characteristics as the companies we studied.

7 Conclusion

The role of software product management is key for building, implementing and managing software products. However, few studies explore how this role is rapidly changing due to digitalization and digital transformation. In this paper, we study how key trends such as DevOps, data and artificial intelligence, and digital ecosystems are fundamentally changing current SPM practices. To support this change, and to provide guidelines for future practices, we identify the key challenges that software-intensive embedded systems companies experience with regards to current SPM practices. Second, we present an empirically derived framework for strategic digital product management (SPM4AI) in which we outline what we believe are key practices for SPM in the age of AI.

References

1. Armstrong, M., Wright, J.: Two-sided markets, competitive bottlenecks and exclusive contracts. Econ. Theor. **32**, 353–380 (2007)
2. Bekkers, W., van de Weerd, I., Spruit, M., Brinkkemper, S.: A framework for process improvement in software product management. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 1–12. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15666-3_1
3. Benbasat, I., Goldstein, D.K., Mead, M.: The case research strategy in studies of information systems. MIS quarterly, pp. 369–386 (1987)
4. Bosch, J.: Speed, data, and ecosystems: the future of software engineering. IEEE Softw. **33**(1), 82–88 (2015)
5. Bosch, J., Olsson, H.H.: Digital for real: a multicase study on the digital transformation of companies in the embedded systems domain. J. Softw. Evol. Process **33**(5), e2333 (2021)
6. Bosch, J., Olsson, H.H., Crnkovic, I.: It takes three to tango: Requirement, outcome/data, and AI driven development. In: SiBW, pp. 177–192 (2018)
7. Ebert, C.: The impacts of software product management. J. Syst. Softw. **80**(6), 850–861 (2007)
8. Ebert, C.: Software product management. IEEE Softw. **31**(3), 21–24 (2014)
9. Ebert, C., Gallardo, G., Hernantes, J., Serrano, N.: DevOps. IEEE Softw. **33**(3), 94–100 (2016)
10. Franch, X., Ruhe, G.: Software release planning. In: Proceedings of the 38th International Conference on Software Engineering Companion, pp. 894–895 (2016)
11. Fricker, S.A.: Software product management. Software for People: Fundamentals, trends and best practices, pp. 53–81 (2012)
12. Gorchels, L.: The Product Manager's Handbook 4/E. McGraw Hill Professional (2011)

13. Gorschek, T., Gomes, A., Pettersson, A., Torkar, R.: Introduction of a process maturity model for market-driven product management and requirements engineering. J. Softw. Evol. Process **24**(1), 83–113 (2012)
14. John, M.M., Olsson, H.H., Bosch, J.: Towards an AI-driven business development framework: a multi-case study. J. Softw. Evol. Process **35**(6), e2432 (2023)
15. Kittlaus, H.B., Fricker, S.A.: Software product management, vol. 298. SpringerVerlag GmbH Germany, Berlin (2017)
16. Koch, M., Krohmer, D., Naab, M., Rost, D., Trapp, M.: A matter of definition: criteria for digital ecosystems. Digit. Bus. **2**(2), 100027 (2022)
17. Krogstie, J.: Modeling of digital ecosystems: challenges and opportunities. In: Camarinha-Matos, L.M., Xu, L., Afsarmanesh, H. (eds.) PRO-VE 2012. IAICT, vol. 380, pp. 137–145. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32775-9_14
18. Leite, L., Rocha, C., Kon, F., Milojicic, D., Meirelles, P.: A survey of DevOps concepts and challenges. ACM Comput. Surv. (CSUR) **52**(6), 1–35 (2019)
19. Liu, Y., Bosch, J., Olsson, H.H., Lantz, J.: An architecture for enabling a/b experiments in automotive embedded software. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 992–997. IEEE (2021)
20. Liu, Y., Mattos, D.I., Bosch, J., Olsson, H.H., Lantz, J.: Size matters? Or not: A/B testing with limited sample in automotive embedded software. In: 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 300–307. IEEE (2021)
21. Maglyas, A., Nikula, U., Smolander, K.: What do we know about software product management? - a systematic mapping study. In: 2011 Fifth International Workshop on Software Product Management (IWSPM), pp. 26–35 (2011). https://doi.org/10.1109/IWSPM.2011.6046201
22. Maxwell, J.A.: Qualitative Research Design: An Interactive Approach. Sage Publications, California (2012)
23. Olsson, H., Sandberg, A., Bosch, J., Alahyari, H.: Scale and responsiveness in large-scale software development. IEEE Softw. **31**(5), 87–93 (2013)
24. Olsson, H.H., Bosch, J.: The HYPEX model: from opinions to data-driven software development. In: Bosch, J. (ed.) Continuous Software Engineering, pp. 155–164. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11283-1_13
25. Olsson, H.H., Bosch, J.: Going digital: disruption and transformation in software-intensive embedded systems ecosystems. J. Softw. Evol. Process **32**(6), e2249 (2020)
26. Olsson, H.H., Bosch, J.: Living in a pink cloud or fighting a whack-a-mole? on the creation of recurring revenue streams in the embedded systems domain. In: 2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 161–168. IEEE (2022)
27. Parikh, N.A.: Empowering business transformation: the positive impact and ethical considerations of generative AI in software product management-a systematic literature review. arXiv preprint arXiv:2306.04605 (2023)
28. Pattyn, F.: Enhancing startup success rates: towards a pragmatic framework for product managers (PFPM). In: IEEE 31st International Requirements Engineering Conference (RE) (2023)
29. Putta, A., Paasivaara, M., Lassenius, C.: Benefits and challenges of adopting the scaled agile framework (SAFe): preliminary results from a multivocal literature review. In: Kuhrmann, M., et al. (eds.) PROFES 2018. LNCS, vol. 11271, pp. 334–351. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03673-7_24

30. Rautiainen, K., Lassenius, C., Sulonen, R.: 4CC: a framework for managing software product development. Eng. Manag. J. **14**(2), 27–32 (2002)
31. Rochet, J.C., Tirole, J.: Two-sided markets: an overview. Institut d'Economie Industrielle Working Paper, pp. 1–44 (2004)
32. Stallinger, F., Neumann, R.: From software to software system products: an add-on process reference model for enhancing ISO/IEC 12207 with product management and system-level reuse. In: 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, pp. 307–314 (2012). https://doi.org/10.1109/SEAA.2012.41
33. Svahnberg, M., Gorschek, T., Feldt, R., Torkar, R., Saleem, S.B., Shafique, M.U.: A systematic review on strategic release planning models. Inf. Softw. Technol. **52**(3), 237–248 (2010)
34. Walsham, G.: Interpretive case studies in is research: nature and method. Eur. J. Inf. Syst. **4**(2), 74–81 (1995)
35. van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., Bijlsma, L.: Towards a reference framework for software product management. In: 14th IEEE International Requirements Engineering Conference (RE 2006), pp. 319–322 (2006). https://doi.org/10.1109/RE.2006.66
36. Wohlin, C., Aurum, A.: Criteria for selecting software requirements to create product value: an industrial empirical study. In: Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., Grünbacher, P. (eds.) Value-Based Software Engineering, pp. 179–200. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-29263-2_9
37. Zhang, H., Li, J., Qi, Z., Aronsson, A., Bosch, J., Olsson, H.H.: Deep reinforcement learning for multiple agents in a decentralized architecture: a case study in the telecommunication domain. In: 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C), pp. 183–186. IEEE (2023)