Antti Paajoki

BEST PRACTICES FOR AND BENEFITS FROM IMPLEMENTING ISPMA'S SPM FRAMEWORK

UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF INFORMATION TECHNOLOGY
2020

ABSTRACT

Paajoki, Antti
Best practices for and benefits from implementing ISPMA's SPM framework
Jyväskylä: University of Jyväskylä, 2020, 138 pp.
Information Systems Science, Master's Thesis
Supervisor(s): Marttiin, Pentti & Salo, Markus

Software has become an integral part of rapidly growing variety of products used in our society every day. The software product market is estimated to be generating hundreds of billions of dollars every year. Software products have several characteristics, which make them unique from physical goods and need to be considered when managing them. Software Product Management practices address that challenge, but the actual state of practice based on prior research shows that there is much to improve. Several different product management frameworks exist for that purpose, one of them being ISPMA's SPM framework. This study investigated the best practices for and benefits from implementing software product management processes based on the ISPMA's SPM framework via a mixed-methods research approach. The combination of quantitative survey and qualitative semi-structured interview research methods was utilized in the empirical part of the study to create an understanding of what the best practices and benefits are. The findings, based on the data collected from an international group of software product management professionals, show that there are several best practices associated with implementing and improving processes based on the ISPMA's SPM Framework. The best practices are associated to three success factors for process improvement: management commitment, roles and responsibilities, and staff involvement. Such best practices include for example mapping current state against the framework, aligned objectives for process improvement, and regular and open communication during the process improvements. Furthermore, the study found various benefits related to such process improvements, which have positive impacts to organizations both internally and externally. The found benefits include for example improvements in communication, customer satisfaction, and employee satisfaction. These results of the thesis contribute to both theory and practice. The research provides insight into what kind of approaches should be taken when improving software product management processes with the support of the ISPMA's SPM Framework, and what kind of benefits can be achievable of such efforts.

Keywords: product management, software product management, process improvement, best practices, benefits, ISPMA, SPM framework

TIIVISTELMÄ

Paajoki, Antti
ISPMA:n SPM-tuotehallintaviitekehyksen jalkauttamisen parhaat käytänteet ja hyödyt
Jyväskylä: Jyväskylän yliopisto, 2020, 138 s.
Tietojärjestelmätiede, pro gradu -tutkielma
Ohjaajat: Marttiin, Pentti & Salo, Markus

Ohjelmistoista on tullut olennainen osa nopeasti kasvavaa joukkoa erilaisia tuotteita, joita yhteiskunnassamme käytetään päivittäin. Ohjelmistotuotemarkkinoiden vuosittaisen arvon arvioidaan olevan satoja miljardeja dollareita. Ohjelmistotuotteilla on useita ominaisuuksia, jotka tekevät niistä uniikkeja suhteessa fyysisiin hyödykkeisiin ja tuotteisiin. Ohjelmistotuotehallinnan käytänteet vastaavat tähän haasteeseen, mutta käytännössä asiassa on paljon parannettavaa. Tarkoitusta varten on kehitetty useita ohjelmistotuotehallinnan viitekehyksiä, joista yksi on ISPMA:n SPM-viitekehys. Tässä tutkimuksessa perehdyttiin kyseisen viitekehyksen soveltamisen parhaisiin käytänteisiin ja viitekehyksestä saataviin hyötyihin monimenetelmätutkimuksen kautta. Tutkimuksen empiirisessä osassa yhdisteltiin määrällisen kyselytutkimuksen ja laadullisen puolistrukturoidun haastattelun menetelmiä kuvan luomiseksi parhaista käytänteistä ja hyödyistä. Tutkimuksen tulokset perustuvat aineistoon, jota kerättiin kansainväliseltä joukolta ohjelmistotuotehallinnan ammattilaisia. Tulokset näyttävät, että ISPMA:n SPM-viitekehyksen avulla prosessien jalkauttamiseen tai parantamiseen on olemassa useita parhaita käytänteitä. Parhaat käytänteet liittyvät kolmeen prosessikehityksen onnistumistekijään: johdon sitoutumiseen, rooleihin ja vastuihin, sekä henkilöstön osallistamiseen. Parhaita käytänteitä ovat esimerkiksi nykytilan kartoittaminen viitekehystä vasten, samansuuntaiset prosessikehityksen tavoitteet sekä toistuva ja avoin viestintä prosessikehityksen aikana. Lisäksi tutkimuksessa löydettiin useita prosessien kehittämiseen liittyviä hyötyjä, joilla on positiivisia vaikutuksia organisaatioille sekä sisäisesti että ulkoisesti. Löydetyt hyödyt sisältävät esimerkiksi parannuksia viestinnässä, asiakastyytyväisyydessä sekä henkilöstön tyytyväisyydessä. Tutkimuksen tuloksista on hyötyjä sekä käytännön että teorian kannalta. Tutkimus tuottaa tietoa siitä, millaisia lähestymistapoja kannattaisi ottaa ohjelmistotuotehallinnan prosessien kehittämiseen ISPMA:n SPM-viitekehyksen avulla, ja millaisia hyötyjä sellaisilla kehityspanostuksilla voidaan saavuttaa.

Asiasanat: tuotehallinta, ohjelmistotuotehallinta, prosessikehitys, parhaat käytänteet, hyödyt, ISPMA, SPM-viitekehys

FIGURES

TABLES

TABLE OF CONTENTS

1 INTRODUCTION

Software can be found in a rapidly growing variety of products, in social, business, and domestic human–machine interactive systems (Xu & Brinkkemper, 2007). Since the inception of software and first software products in the 1960's (Xu & Brinkkemper), increasing number of businesses have started moving from developing tailor-made software to software products in the last decades (Artz, van de Weerd, Brinkkemper & Fieggen, 2010). During this period, the business around software products has developed into an enterprise software market with an estimated (Gartner, 2019a) worldwide spending volume of $457 billion in 2019 and expected growth rate of 10,9% for 2020.

Software products, which are "products whose primary component is software" (Kittlaus & Fricker, 2017), have several characteristics that make them unique from physical goods which need to be considered in managing them (Kittlaus & Fricker, 2017). However, product management as a way of connecting software development and business – and the tasks and processes related to those – have not been defined in large-scale before the 2000s (Mkrtchya, 2019, Fricker, 2012). Historically, there has been little education on software product management available, except for marketing and sales (van de Weerd, Bekkers & Brinkkemper, 2010a). Studies show that by implementing software product management practices the product quality increases by 80% and time-to-market is reduced by 36% (Ebert, 2007, Ebert & Brinkkemper, 2014). Additionally, delays are reduced up to 85% (Ebert & Brinkkemper, 2014). However, while 70% of enterprises see that a software product manager has the overall responsibility of the product success, only 30% of the enterprises work according to their own vision (Ebert, 2018). According to Fricker (2012) and Maglyas et al. (2017), there has been several different attempts to create a suitable framework for software product management.

SPMBoK (Software Product Management Body of Knowledge) is one of the most comprehensive product management approaches specializing to software product management. It is based on a framework, a curriculum, and a glossary. The framework, called ISPMA's SPM Framework, ties in together core areas such as product strategy and product planning, and supporting areas re-

lated to the discipline of software product management. It has been developed since 2009 by the International Software Product Management Association (ISPMA), and it is based on best practices from the enterprises and academic research. However, the best practices of applying the framework into enterprises and benefits gained by doing so has not been studied in academics. (SPM Body of Knowledge, 2019)

Software Process Improvement (SPI) is a popular approach to deliver improvements in software products and related processes (Humphrey, 1989, Kuhrmann et al., 2016). Many companies have executed either formal or informal SPI programs, which have delivered significant benefits (Hall et al., 2002). Best practices and success factors for process improvement in software product management, however, have been a relatively low focus area until recently (Ebert & Brinkkemper, 2014, Kuhrmann et al., 2016).

There is a need for understanding what the best practices are for implementing or improving software product management processes, and what benefits can be gained by doing so. To address this research gap, the following research questions have been formulated:

RQ1: What best practices have been used in different organisations in applying the ISPMA's SPM framework?
RQ2: What benefits have been achieved by applying the ISPMA's SPM framework?

A mixed-methods research approach, combining quantitative survey and qualitative semi-structured interviews, was used to address the research questions. The research was done in the context of ISPMA's certified members, a group of international software product management professionals with varying backgrounds.

The research results show that there are several best practices associated with implementing and improving processes based on the ISPMA's SPM Framework. Such practices include mapping current state against the framework, aligned objectives for process improvement, regular and open communication, active involvement of management in the process, focusing to clear role definitions, and running an expert-led community of practice. The benefits associated to improving processes based on the framework according to the study include increased awareness of software product management, improved employee satisfaction, improved communication, less defects in products, increased customer value and, improved customer satisfaction.

The findings of the research contribute to the overall knowledge about best practices for and benefits from implementing or improving software product management practices. These apply specifically in the context of the ISPMA's SPM Framework, for which such studies have not been done earlier.

The remainder of this thesis is structured as follows: Chapter 2 explains the concepts of product, software product and software product management based on prior research. In addition, various product management frameworks are compared concerning their suitability to software product management. Chapter 3 then introduces the ISPMA's SPM Framework in details. Chapter 4

investigates process improvement in software product management and presents potential best practices for it. Chapter 5 describes the research strategy to address the research questions and research methods used in the empirical study. Chapter 6 describes the results of the mixed-method research utilizing quantitative survey and qualitative interview approach. Finally, Chapter 7 summarises the thesis by answering the research questions RQ1 and RQ2, draws conclusions and implications to theory and practice, discusses limitations, and outlines areas for future research.

2 PRODUCT MANAGEMENT

To understand software product management, it is first necessary to understand what software products are. This chapter begins by introducing the generic concept of a product and product management. Then, software products and their unique characteristics compared to other products are reviewed, followed by differentiation of software products from tailor-made software. Software product management is then introduced with the most typical organisational role associated with it: the product manager. After review of common challenges during the software product life cycle, the chapter is closed with description and comparison of the most used software product management frameworks.

2.1 Product

Widely accepted and known definition of a product is made by Kotler and Armstrong in their book Principles of Marketing (1994, pp. 432). They define product "as anything that can be offered to a market for attention, acquisition, use or consumption that might satisfy a want or need". Later, the definition has been clarified to include more than just tangible objects – for example services, which consist of activities and benefits offered for sale, which do not result in ownership of anything (Kotler, Armstrong, Harris & Piercy, 2017). Kittlaus and Clough (2009, pp. 6) have defined further, that a product is "a combination of (material and/or intangible) goods and services, which one party (called vendor) combines in support of their commercial interests, to transfer defined rights to a second party (called customer)". This definition also includes the customer-supplier relationship, while the traditional marketing definition addresses bigger markets (Kittlaus & Fricker, 2017).

Products are a key element in the market offering of an enterprise, which targets to build profitable customer relationships. Key to that relationship is to bring value to the customers, for which Kotler et al. (2017) have introduced

three levels of product, each adding more customer value as shown below (figure 1).

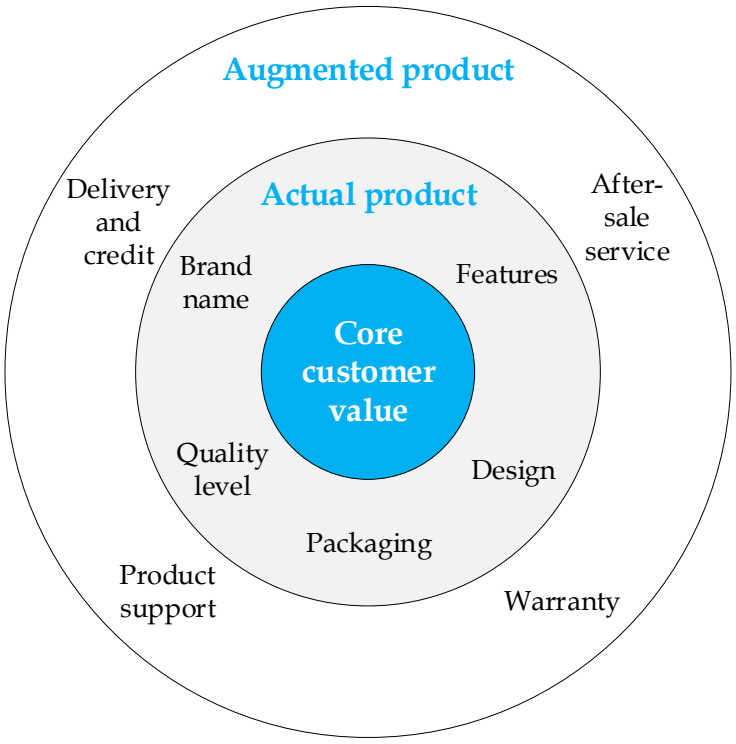

FIGURE 1 Three levels of product (Adapted from Kotler et al., 2017, pp. 228)

The first and the most basic level is core customer value, which answers the question of what the customer is really buying. In most cases, it is intangible, and the customers are seeking benefits that fulfil specific needs or wants. The second level of the product is the actual product, which contains product and service features, design, brand name, quality level, and packaging. These are carefully combined to deliver the core customer value and to differentiate from the competition. The third and final level is the augmented product, which extends the core customer value and the actual product by offering additional services and benefits to satisfy the customer further. Such product properties are delivery, product support, after-sale service, and warranty. (Kotler et al., 2017)

The three-level product by Kotler et al. (2017) has similarities to another famous concept: the "whole product". It was first introduced by Regis McKenna and later popularized by Geoffrey Moore in his bestseller "Crossing the Chasm" (1991). Moore (1991) defines it as "the minimum set of products and services needed to fulfil the compelling reason to buy for the target customer", and later (Moore, 1994) as "the complete suite of products and services necessary to fulfil the promised value proposition". The whole product must be created "by thinking through your customer's problems – and solutions – in their entirety" and it includes "the core product plus everything else you need to

achieve your compelling reason to buy", which "may be provided in-house or by using partners and alliances". (Moore, 1991)

Within most companies there is a hierarchy of different products. A single product may be a part of another product or product line, packaged with other products (bundle), or offered as a more extensive solution or system to meet broader sets of customer needs. The whole product offering is usually part of a larger product portfolio. Additionally, products can be broken down into smaller product elements, which can be either tangible or intangible. Products can be furthermore built utilizing product platforms or architectures, so that a higher degree of standardisation can be achieved across the portfolio. This hierarchy of products is visualized below (figure 2). (Haines, 2009)

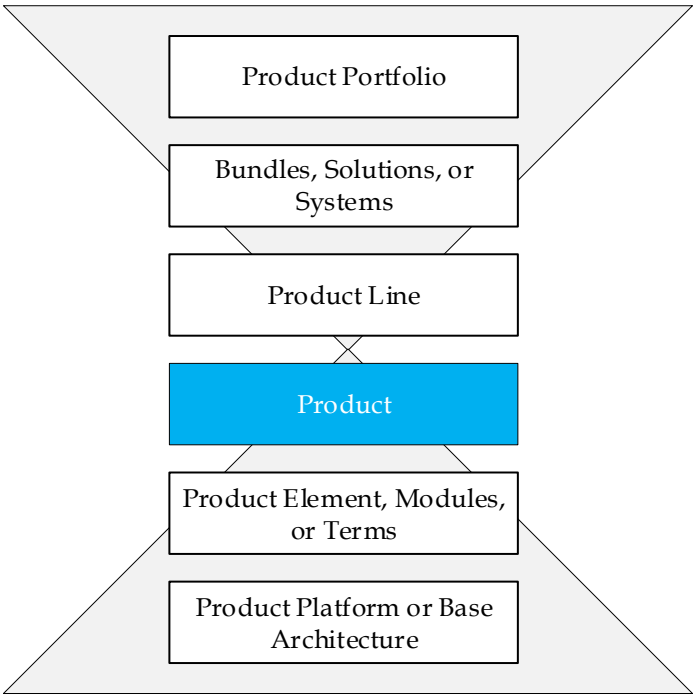

FIGURE 2 Typical hierarchy of products (Adapted from Haines, 2009, pp. 7)

According to Ebert (2018), products need to be increasingly more solution-like, and the solutions need to include more and more services and integration of business processes. In the end, the customers are not interested in features, but in satisfying their needs. A clear trend is also, that the customers are more integrated part of the value creation process, and they are more interested about the quality and sustainable supply networks for the products (Ebert, 2018).

Products and services also fall into two main categories based on the types of consumers that use them: consumer products (business-to-consumer, B2C) and industrial products (business-to-business, B2B). Consumer products are bought by end customers for personal use, while industrial products are bought by enterprises for further processing or for conducting business. Therefore, the main difference from this viewpoint is the purpose for which the product is purchased. (Haines, 2009)

2.2 Product management

According to Gorchels (2011), Haines (2009) and Chisa (2014), the earliest forms of Product Management were conceived at Procter & Gamble in the 1930s to improve the oversight of their expanding consumer products business. It all started when an advertising manager named Neil McElroy proposed to the executive team at Procter & Gamble an role of a "Brand Man", who would be responsible for a product instead of a business function (Chisa, 2014). According to Gorchels (2011), this was implemented at Procter & Gamble with two brands of soaps, and the approach became so successful, that the practice was copied by most consumer products companies. The job definition for the role had many similarities to modern-day product management tasks (Chisa, 2014).

The word "management" is derived from Latin: *manu agere*, or "to lead by the hand". Most definitions of management generally include the usual cycle of business elements: setting goals, directing human and financial resources, assessing outcomes, and reassessing and/or resetting goals (Haines, 2009). Ebert and Brinkkemper (2014, pp. 1) define product management as the "discipline and business process which governs a product from its inception to the market or customer delivery and service in order to generate biggest possible value to the business".

Product management as a business process covers and provides leadership to different aspects of the product during its lifecycle (Ebert & Brinkkemper, 2014). According to Ebert & Brinkkemper (2014), the main phases of the product lifecycle are "Strategy", "Concept", "Market Entry / Development" and "Evolution". The model matches well a generally accepted product management life cycle model by Haines (2009), where the four areas of work are "Discovery and Innovation", "New Product Planning", "New Product Introduction", and "Post-Launch Product Management", which is shown below (figure 3).

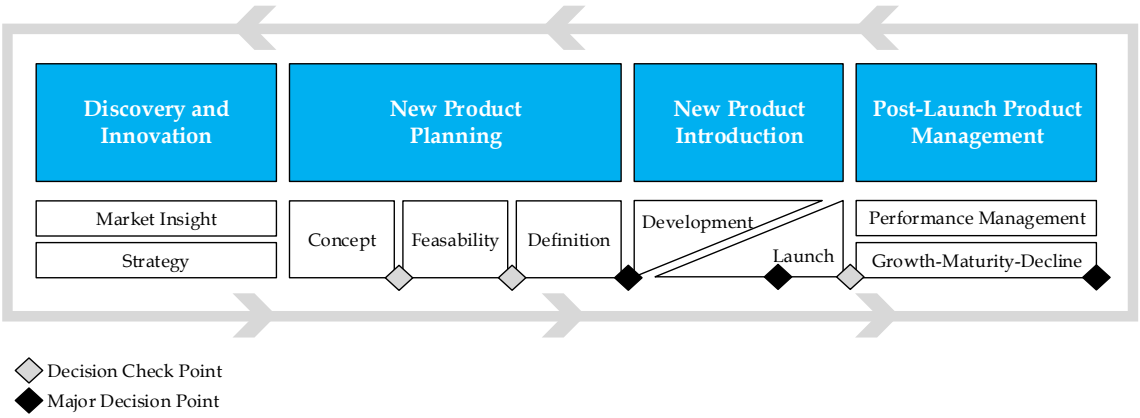

FIGURE 3 Product Management Life Cycle Model (Adapted from Haines, 2009, pp. 21)

The model is a simplification – a linear, progressive, and static description of something that is three-dimensional, recursive, and dynamic in nature. Underneath the model, marked with diamond shapes there is also depicted a standard phase-gate product development process, or as some refer to it, the "New Product Development (NPD)" process. Companies trying to improve time-to-market have utilized phase-gate processes for decades. (Haines, 2009)

The model also incorporates parts of well-known product life cycle models (Fox, 1973, Anderson & Zeithaml, 1984 etc.) in the two last major life cycle phases. The traditional product life cycle theory is defined by a pattern of sales against time, which is generally visualized as a bell-like shaped curve, where the stages most typically are introduction, growth, maturity, and decline. Introduction is where the product lifetime is only at its beginning – Haines (2009) sets the growth, maturity, and decline stages into the Post-Launch Product Management life cycle phase.

The product is positioned differently in the company's product portfolio during its lifetime. It requires initial investments in the product definition phase, where the concepts and feasibility are evaluated. The need for investments grows during the introduction phase, in which the focus is on development, launch, and the following growth stage. During maturity and decline the product acts as a revenue source for other products before it is eliminated. Each phase requires dedicated product management activities, and performance and risk management need to be ensured in all these three stages. (Fricker, 2012)

The models highlight well the differences between managing a project and managing a product. The difference is focus: project focus is on delivering one specific deliverable in time, budget and quality, while management of the product means looking at it from overall market success and lifetime point-of-view, considering its subsequent releases and related services. (Ebert & Brinkkemper, 2014)

In product management, the person in charge who leads and manages the process is the product manager, who masters and owns the life-cycle processes (Haines, 2009, Ebert & Brinkkemper, 2014). The product manager must balance between projects, people, and politics (Ebert & Brinkkemper, 2014). His main tools for the job are roadmaps, business cases, requirements, and milestone reviews (Ebert & Brinkkemper, 2014). According to Springer and Miler (2018), the most important skills for a product manager, regardless of the business environment, are highly developed soft skills, basic hard skills, knowledge of different techniques and tools, and empathy towards the customers and users.

Product managers have been sometimes dubbed as "CEO of product" (Horowitz, 1998) or "mini CEO" (Ebert, 2007). According to Steinhardt (2010) and Maglyas, Nikula and Smolander (2013), such definition may have distorted the view of what product managers really do. Firstly, "being labeled or treated as a product CEO can be a daunting situation, since it nearly always means operating without the authority and resources available to a corporate CEO" (Steinhardt, 2010, pp. 19). Secondly, the product manager role varies between expert, strategist, leader, and problem solver profiles – being a "mini CEO"

would require excelling in all aspects, while in most cases the responsibilities are split into multiple roles within an organisation (Maglyas et al., 2013).

Considering the multifaceted definition of the product described earlier, a one-to-one relationship between product and product manager cannot be assumed. Depending on the organisation, a product manager can be in fact partly or wholly responsible for any part of the product hierarchy (Haines, 2009). According to Product Focus (2019), product managers were on average responsible of five products based on a survey done to 1174 people in 46 countries and 876 companies.

Many failed companies have stated the lack of product management as a significant weakness. Poor product management leads to overemphasizing technology instead of customer value, and to unsuccessful product development due to bad project planning and change management. (Ebert, 2014)

Ebert (2018) has further found that 70% of companies see that product managers have the overall responsibility for the success of the product, but only 30% were actively implementing their own vision. The number of companies having that vision has gone up from 53% in four years, while the number of companies practicing so has worryingly gone slightly down from 33% (Ebert & Brinkkemper, 2014).

According to Ebert (2018), current product management trends include increasingly increasing challenges for the product manager. There is a trend on focusing even more on the value and solution, rather than being technology driven. One must think of the products as more holistic solutions including services and ensure that the customers are part of the value creation. The products must fulfil the increasing quality and sustainability demands, which overarch all the way to the partner and supplier networks. The knowledge management of the constantly changing markets, customers, products, and technologies must be effective and support the people-dependent product management processes requiring agility to manage the uncertainties. (Ebert, 2018)

2.3 Software product

The word “software” itself was coined by statistician John Tukey in an article from AT&T’s Bell Labs in 1958 (Abran & Moore, 2004, Kittlaus & Fricker 2017). In the early days of computing and software development, all software which was not provided alongside the computer had to be tailor-made for the purpose.

Software was not considered as independent and separate before the early 1960, which was also the same time entrepreneurs started to see an opportunity for software products business (Kittlaus & Fricker, 2017). First software products were developed because of agreement between IBM and United States Department of Justice to unbundle software from hardware (Xu & Brinkkemper, 2007). The announcement about it by IBM in 1969 can be viewed as a birthdate for the software industry as we know it (Kittlaus & Fricker, 2017). During the last decades, more and more businesses have started moving from developing

tailor-made software to software products (Artz, van de Weerd, Brinkkemper & Fieggen, 2010), following the phases of evolution of the software market domain by Tyrväinen, Warsta and Seppänen (2008). Within this period, the business around software products has developed into an enterprise software market with an estimated (Gartner, 2019a) worldwide spending volume of $457 billion in 2019 and expected growth rate of 10,9% for 2020.

Software can be found in a rapidly growing variety of products. According to classification of software by Xu and Brinkkemper (2007), software can be classified by how it is sold (appliances or software) and how many copies of it exist (one or many). The classification is shown in the figure below (figure 4).

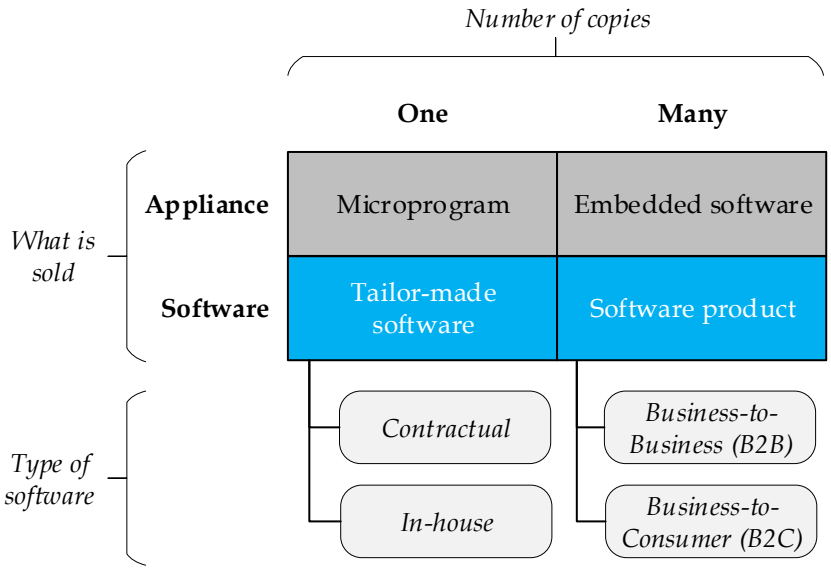

FIGURE 4 Classification of software (Adapted from Xu and Brinkkemper, 2007, pp. 532)

Appliances are not considered as software offerings at all since the product being sold is the appliance. The software within appliances are either microprograms, which are non-copiable one-time applications, or embedded software, when the appliance is sold in multitudes (Xu and Brinkkemper, 2007). Kittlaus and Fricker (2017, pp. 11) further define, that embedded software is "software parts of software-intensive systems that are not marketed and priced as separate entities". All components together enable such technical system to become a product (Kittlaus & Fricker, 2017).

Software also exists in different multitudes. Software can be either tailor-made software or product software (software product). Tailor-made software is done by a software project vendor for one buyer as contractual software, or for one internal organisation to improve efficiency (Xu and Brinkkemper, 2007). There are several aspects of how software product differs from tailor-made software. The main difference between tailor-made software (between vendor and customer) and market-driven software product (between vendor and market) is that with software products, the product development cost and vendor

revenues, are not tied to a specific customer but to a larger market, consisting of any number of potential buyers (Gorschek, Gomes, Pettersson & Torkar, 2012).

Considering the development of the software product, more differences compared to tailor-made software can be found in development life cycles, requirements, architecture, implementation, release management and delivery. By utilizing a suitable development life cycle, it is possible to shorten the release cycles, improve software product quality and time-to-market for releases and features. Tailor-made software can be considered ready and moved to maintenance when customer approves their requirements to have been fulfilled, but software products are rarely ready, and need to evolve over the lifecycle with changing demands from many sources. Therefore, it is crucial to define a sustainable long-term architecture for a software product, which can easily be adapted and re-purposed when facing changes in requirements, technical platforms, customer environments, and competition. (Xu & Brinkkemper, 2007)

Software products can be further grouped into business-to-business (B2B) products and business-to-consumer (B2C) products, which both have different typical characteristics (Xu & Brinkkemper, 2007). Such characteristics are for example found in the business model, the split between customer and user, and the criticality of the software for the buyer (Xu & Brinkkemper, 2007, Reeves & gaines, 2018).

A *software product* is defined as "product whose primary component is software" by Kittlaus and Clough (2009, pp. 6). The wording "primary" gives room for flexibility and it is important to define some examples, which therefore are not software products. For example, mobile phones are not considered software products, even when the embedded software is a large and important part of the product (Kittlaus & Fricker, 2017).

It should be also noted that the definition of a product by Kittlaus and Clough (2009, pp. 6) includes a phrase "in support of their commercial interests" refers to business but does not mean that it necessarily leads to payments. There are also commercial interests behind Open Source, and even products which are offered free of charge, such as Adobe's Acrobat Reader, have a commercial goal – to increase market penetration of another paid product within the software vendor's portfolio (Kittlaus & Fricker, 2017).

The usefulness of a software product is defined by the functionality it provides via its different interfaces, and value is generated because of such functionality. Businesses in many industries depend on software products, provided that the generated value is understood by those who are targeted with these software products. (Fricker, 2012)

As opposed to physical goods, there are characteristics in software products, which are different from most other products and have a large impact on what software product management practices should include. Four of such characteristics are described below.

Firstly, the manufacturing and distribution of extra copies of the product does not require, or requires very little, extra costs for the company (van de Weerd, Brinkkemper, Nieuwenhuis, Versendaal & Bijlsma, 2006, Kittlaus &

Fricker, 2017). Software business can yield up to 99% gross profit margins for product sales (Xu & Brinkkemper, 2007), since the manufacturing and distributing costs are negligible compared to the development costs of the software product (Fricker, 2012). Software typically have high fixed development costs, which usually cannot be recovered if a software product is not successful (Xu & Brinkkemper, 2007).

Secondly, software products can be changed or updated relatively easily by using patches or release updates (van de Weerd et al., 2006). This flexibility makes incremental product development possible, which enables a rapid break-even and high return of investment (Fricker, 2012). Flexibility also means possibility of re-configuring and adapting the product to completely new purposes and usage contexts over the lifetime (Kittlaus & Fricker, 2017).

Thirdly, the release frequency is high, since the product can be altered easily (van de Weerd et al., 2006). The frequency of change over the life cycle of the software emphasizes importance for requirements management and organisation (van de Weerd et al., 2006, Kittlaus & Fricker, 2017). This is especially valid for large and complex software products that offer integration with other software (Fricker, 2012).

Fourthly, the software product business is not without risks. Developing software includes risks related to process maturity, technological newness, development team, application size, and complexity (Benaroch & Appari, 2010). The productivity of the "best" employee and "worst" employee frequently have up to 10- or 20-times difference, and about 75–80% of the product development projects are late and over budget (Xu & Brinkkemper, 2007). Risks are especially high in software start-ups, where the aim is to create innovative high-tech products, and to aggressively grow to scalable markets with little operating history (Giardino, Paternoster, Unterkalmsteiner, Gorscheck & Abrahamsson, 2016). Vast majority of such start-ups fails within two years of their creation, primarily due to self-destruction than competition (Giardino et al., 2016).

2.4 Software product management

Many software companies focus too much on projects, technology and features instead of value, market understanding and products (Ebert & Brinkkemper, 2014). The effects are depressing – according to Ebert and Brinkkemper (2014), the average of delivering new products on time is as low as 51%, and only 52% of original requirements appear in product releases. One of the main reasons for failure are dysfunctional organisations without defined product owner with business success accountability (Ebert & Brinkkemper, 2014).

The answer to this is software product management (SPM), which can be considered as a combination of many different disciplines (Maglyas et al., 2017), including marketing, development, and corporate strategy among other company functions. Software product management contains multitude of activities, which are to be performed over the non-predetermined lifecycle of the software

product, from idea to retirement (Kittlaus & Fricker, 2017). This section goes deeper into the software product management by looking at the software business models, key activities and roles, and various challenges related specifically to software product management.

2.4.1 Software business models

Being successful in a fast-moving digital economy – such as the software industry – highly depends on the business model. According to Valtakoski and Rönkkö (2010), the fundamental idea for a business model is linked to Richard Normann's book "Management for Growth" from 1977, but the concept really gained popularity in the 1990s, when it was commonly used to describe the business logic of start-up companies during the dot-com boom. Since then, the concept of business models has been used in research related to mainstream management concerning strategic management, innovation, entrepreneurship, and information systems (Valtakoski & Rönkkö, 2010).

Business model in its essence reflects how a company plans to make money (Kittlaus & Fricker, 2012). The definition of the term business model, however, is subject to debate and related terms such as strategy and revenue model are often used interchangeably, leading to confusion (Valtakoski & Rönkkö, 2010, Schief & Buxmann, 2012). For example, Schief and Buxmann (2012) have created an extensive framework for software industry business model, which consists of 20 business model elements clustered into five groups: strategy, revenue, upstream downstream and usage. Furthermore, for each element, there are 2 to 11 different options to choose from. The model highlights well the complexity of software business, and the range of different business models in the software industry, which must be considered when planning the direction of the companies and their strategies.

Popp and Meyer (2010) have defined more simply, that the business model consists of a type of products and services provided, the archetype of the business model, and a revenue model. Typical business models for software business are Inventor, IP lessor, IP distributor, Contractor, and Physical lessor. An "Inventor" is the creator of intangible goods, which means creating individual software and selling it to individual customers with the intellectual property (IP) rights. When a software company is creating standard software (software product), they act as an "IP lessor", meaning that they provide temporary rights to use the software via licenses while retaining the intellectual property rights. A software company can also act as an "IP distributor" and provide customers rights to use products of other software vendors. Very commonly, a software vendor also adopts a "Contractor" business model, which may include for example software customization, consultation, or support services. In addition, especially in Software-as-a-Service (SaaS) business, the software vendor may also act as a "Physical lessor" of physical server hosting space and computing resources. The software vendors typically adopt hybrid business models which include one or more business models. (Popp & Meyer, 2010)

Product management has been established in other industrial sectors, especially in manufacturing, already since the industrial revolution in the 19th century (van de Weerd et al., 2006). As stated in the previous section, first software products were established in the 1960's. However, software product management is much younger, growing discipline.

The pioneer within software companies to incorporate product management principles was Intuit, which was founded in 1983. Intuit created a very successful finance software product Quicken for home users, while applying previous experiences of one of the founders – Scott Cook, a former "brand man" from Procter & Gamble. In the 1990's, Microsoft and other software companies followed. (Mkrtchyan, 2019)

The first book on software product management was published in 2002, the first academic workshop of the International Workshop on Software Product Management (IWSPM) took place in 2006, and the first conference on the International Conference on Software Business (ICSOB) took place in 2010 (Fricker, 2012). The knowledge and experience from other business areas is only partially transferable to software (Kittlaus & Fricker, 2017), and there has not been much education on software product management available, except for marketing and sales (van de Weerd, Bekkers & Brinkkemper, 2010a). One of the first university level Master's programmes focusing solely on software product management starts in 2020 at LUT University in Finland (LUT University, 2020).

Software product management applies to both organisations in software industry and other industries. Software industry in this context covers vendors of software products and software-intensive technical services, which may be licensed products, Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), and applications running on various kinds of smart devices (software industry). Software product management is applicable also in other industries such as with vendors of software-intensive products (e.g. cars or smartphones), vendors of professional human services (in which software is used to increase productivity), and corporate IT organisations. (Kittlaus & Fricker, 2017)

Gartner (2018) has found that majority of organisations are moving from project to product delivery. The most recent Gartner (2019b) survey about the subject reports that 85% of organisations favour a product-centric application delivery model. The benefits from moving from project to product-centric digital business models include faster business outcomes, improved customer experience, more flexibility, reduced friction, and common goals within the organization (Gartner, 2018).

In the 2019 study (Gartner, 2019b), IT and IT-business professionals reported that a product-centric model was utilized in 40% of their work, and Gartner predicts the proportion to rise to 80% by 2022. Time-to-market was reported as the main driver to this transformation alongside digital business (digitalization). More than half of the respondents highlighted culture clash between "the business" and "information technology" as a top challenge, and 46% have already appointed product managers within their organisation.

2.4.2 Key activities and roles

According to recent study by Maglyas, Nikula, Smolander, and Fricker (2017), the core software product management activities are vision creation, product lifecycle management, roadmapping, release planning, and product requirements engineering. Additionally, there are six supporting activities: strategic planning, portfolio management, product analysis, product launches, product support, and product development. The table below (table 1) groups the activities according to the type of the activity and gives an abbreviation for each activity which are used later in this study.

TABLE 1 Key software product management activities (Adapted from Maglyas et al., 2017, pp. 36)

Activity type	**Activity**	**Abbreviation**
Core	Vision creation	VC
	Product lifecycle management	PLM
	Roadmapping	RM
	Release planning	RP
	Product requirements engineering	RE
Supporting	Strategic planning	SP
	Portfolio management	PM
	Product analysis	PA
	Product launches	PL
	Product support	PS
	Product development	PD

The core software product management activities represent the main responsibilities for the product management, which are found to be common across different companies. The core activities play a critical role in developing and releasing a product to the market successfully, and it is reasonable to expect that every software product management professional is familiar with the core activities. (Maglyas et al., 2017).

The product management is partially responsible or participating to supporting activities, which contribute to the product success. The degree and implementation of these activities vary between companies and the personal skills of product management professionals vary depending of their background, business domain and the product type. (Maglyas et al., 2017).

Like with other products, the product manager is the key role concerning different activities around the software product. In addition to generic product manager role discussed earlier, there has been academic research made on the role of a *Software Product Manager*. Maglyas et al. (2013) have developed a Software Product Management Roles Framework, which enables product managers

and top management to understand the tasks required from software product managers. The software product manager role may vary from being an expert, strategist, leader or problem solver – or the all-encompassing "mini-CEO" (Maglyas et al., 2013).

Springer and Miler (2018) have further defined eight different software product management personas, depending of the size of the organisation and the type of the product (B2B or B2C). Their study also produced an archetype for a "generic" software product manager based on the common attributes, which is presented below in a tabular form (table 2) (Springer & Miler, 2018).

TABLE 2 Archetype of a Software Product Manager (Springer & Miler, 2018)

Attribute	**Description**
Objectives	Achieving goals by implementing the product strategy and consistent product vision.
Responsibilities	Defining goals, proposing solutions, prioritizing projects or tasks, user research, analysis of requirements, market analysis, stakeholder management, cooperation with the development team.
Main competences	*Soft skills*: communication, negotiations, teamwork, decision-making, curiosity of the world, open-mindedness, assertiveness, understanding of human behaviour, inquisitiveness, networking, leadership predispositions, consistency, and perseverance. *Hard skills*: ability to understand the problem domain, data analysis and synthesis, knowledge of business analysis and project management, interface prototyping, willingness to learn.
Cooperation with other teams	Cooperate with all product stakeholders and development team.
Techniques	Techniques that support: verifying a product vision and strategy, product delivery, user research.
Tools	Tools that support: task and backlog management, data analysis, user research, documentation, prototyping, remote cooperation.

One product manager is rarely responsible for all software product management activities (Maglyas et al., 2013), and as described earlier, the product manager may be responsible for one or more products (Haines, 2009, Product Focus, 2019). For software product managers, this can also mean that the product is not "just" software. According to Popp and Meyer (2010) though most products in the software industry are intangible (e.g. software and other intellectual property), almost all company offerings also offer human services (people's time and effort), or other types of products, such as financial products (e.g. cash or other assets), or physical products (e.g. "real" physical products and goods). The products can be offered separately or considered as bundles (Kittlaus & Fricker, 2017). According to Product Focus (2019), 57% of product people are responsible for a mix of different types of products, and 18% manage propositions made up of software, hardware, and service elements. The most common combination is software and services, which 27% of the product people work with.

While the role of the product manager varies, the role is often confused with the roles of project manager (Manteli, van de Weerd & Brinkkemper, 2010a,

Ebert, 2014) and marketing manager (Maglyas et. al, 2013, Ebert, 2014). Relationship and differences between product and project managers were studied by Manteli et al. (2010), and Ebert (2014) demonstrated differences between all three. In summary, the product manager focuses on results and asks what and how to make it, the marketing manager focuses on the market and asks how to best sell it, and the project manager focuses on the implementation and asks how to best execute the project or contract (Ebert, 2014).

Recently with agile development methodologies, the role of the product manager is further confused with the relatively new role of product owner (Ebert, 2018). Agile development methods build around empowered and self-organizing teams, which coordinate the work themselves and have a strong focus on collaboration and communication (Šmite, Moe & Ågerfalk, 2010). The activities are supported by agile practices such as customer collaboration, stand-ups, reviews, pairing, retrospectives, and the planning game (Šmite et al., 2010). As the agile methodologies are today the most used software development practices (Product Management Festival, 2019, Product Focus, 2019), the agile development, the differences and similarities between product manager and product owner roles, are further detailed in the next section.

2.4.3 Agile methodologies and software product management

In 2001, several developers and experts practicing lightweight software development methods, defined together a shared agile manifesto, including four core values and 12 principles of agile software development (Agile Manifesto, 2001). Agile development methodologies are widely used today – surveys by Product Management Festival (2019) and Product Focus (2019) both report over 90% adoption of some variant of agile development methods. Product Focus (2019) reports 68% of respondents using a mix of approaches, which may be either agile or traditional methods. However, the numbers vary by industry. A recent longitudinal industry research by Ebert (2018) to 1500 decision-makers in worldwide business-to-business (B2B) companies showed much lower agile adoption rate of 55%.

The most used agile method today is Scrum with 54% usage rate according to yearly "state of agile" survey by Collabnet (2019). Other commonly used agile methodologies are Extreme Programming (XP), Kanban, and hybrids of the different methods. One of the issues of agile methods is scaling it across multiple teams in a large organisation, for which multiple approaches have been developed. The most used formalized approach is Scaled Agile Framework (SAFe) with 30% adoption rate in agile adopters, with Scrum of Scrums as the second (Collabnet, 2019).

The wide adoption and success of agile approaches in software development doesn't mean that it suits every scenario, and there are still many projects where, due to contents, people and other factors, different methodological approaches may be appropriate (Kittlaus, 2012). In a business-to-business industry research, Ebert (2018) noted only two companies within hundreds of interview-

ees deploying scalable agile schemes, and one of them stopped using SAFe due to its vast complexity. Additionally, several others were looking towards more light-weight scalable agile techniques.

The guiding document for Scrum is The Scrum Guide by the creators of Scrum. Scrum is not a full-blown method or technique, but more of a guiding framework based on a philosophy, which values self-organisation and the individual abilities of the team members. The latest 2017 edition of the guide contains only 19 pages and leaves out definitions for fundamental terms related to software product management, such as "product" and "release". (Schwaber & Sutherland, 2017)

While being agile is nothing new to a seasoned software product manager, as certain agility has been a prerequisite to achieve product success, the Scrum role definition of "product owner" conflicts with the state-of-the-art software product management. Especially the time usage demands associated with product owner role conflict with other requirements typically imposed to a product manager. In addition, typical product management tasks include significantly more than defined in Scrum. (Kittlaus, 2012)

According to Vähäniitty (2012), there seems to be a disconnect between "traditional" software product management and agile software development. Most of the product management literature sees product development as a straightforward activity, which can be planned in detail and then executed to said plan, which conflicts with the agile software development mindset. On the other hand, most of the literature on agile software development has been focused on a narrow context of a single team working with a single backlog and product – all while the most difficult management questions are "thrown" to the over-burdened product owner (Vähäniitty, 2012).

The role of product owner is also in practice often confused with the agile role product owner, and several studies have pointed out the weakness of overloading the product owner with conflicting tasks and expectations in agile environments (Ebert, 2018). Leffingwell (2011, pp. 280) summarises this well: "Given this (the product manager's) set of responsibilities, it is clear that – even with a staff of competent product owners – product management remains an important function in agile development".

According to Product Management Festival (2019), the ways how different organisations handle these roles vary greatly. 28% of organisations have only product managers, but no product owners. 12% or organisations have only product owners, but no product managers. 32% of the organisation either make no distinction between the two, or both roles are managed by the product managers. And finally, 21% of organisations either have one-to-one or one-to-many relationships between product managers and product owners.

In a similar survey done by Product Focus (2019), only 37% of product managers in companies using agile methodologies are also the product owner, and 23% of respondents, which utilize agile methodologies, have only product managers and no product owners. Both surveys prove that there is high variation on how this challenge is handled.

A mature software organization should be able to select the most optimal method for each development project, and the software product manager should be able to collaborate with the project teams regardless of the chosen development method (Kittlaus, 2012). Regardless how the actual organisation is set up, the roles need to be empowered and held accountable for results, so it can stimulate working motivation and enable faster and more effective decision-making in a company (Ebert, 2014).

2.4.4 Global software development

There is a strong trend for organisations to utilize multisite teams – spanning geography and cultures – to build software. This so-called global software development (GSD) paradigm has become a dominant and is being increasingly adopted either out of necessity, or in search of the benefits provided by the approach. The benefits include access to specialized and diverse resources globally, better use of limited resources, reduced costs, and better competitiveness by accessing a global market. (Vizcaíno, García, Piattini & Beecham, 2016)

Already in 1968 by Melvin Conway, it was noted that the structure of a system or product mirrors the organisation that designs it (Conway, 1968). This has been commonly known since as the Conway's Law, and primarily this is due to the communication needs of the people working with the design (Herbsleb & Grinter, 1999). In multi-site development the coordination mechanisms and informal communication have a key role of overcoming challenges caused by the distance, and there needs to be a common understanding of the development process (Herbsleb & Grinter, 1999).

The pressure to reduce time-to-market and to cut costs motivates software product vendors to adopt global software development (Colomo-Palacios, Soto-Acosta, García-Peñalvo & Carcía-Crespo (2012). In software product management, many aspects are already complex with local teams, but some areas increase in difficulty when adopting global software development approach (Colomo-Palacios et. al, 2012). Such areas are for example release planning (van de Weerd & Katchow, 2009, Colomo-Palacios et. al, 2012), project management (Colomo-Palacios et. al, 2012), and quality management (van de Weerd & Katchow, 2009, Colomo-Palacios et. al, 2012). This brings additional challenges to both product managers and project managers leading the development efforts (Colomo-Palacios et. al, 2012).

The adoption of global software development paradigm, especially with scaled agile practices, can positively impact the capabilities of the company. For example, longitudinal case study by Ramesh and Clear (2018) showed that, in addition to the internal software process improvements caused by such adoption, the outbound adaptive and innovative capabilities of the companies are improved. Such capabilities are critical for a market-driven software company, and for the success of the software product management as well. (Ramesh & Clear, 2018)

2.4.5 Additional challenges

In software product management, there are many additional challenges caused by internal and external factors, which need to be considered when implementing software product management practices. For example, the strategy and size of the company plays a major factor of where efforts should and can be directed, the life cycle state of the market and product must be considered, and finally roles and responsibilities need to be suitable for the purpose. This section details some of those challenges which have an impact on how to setup software product management practices.

In the software markets there can be identified four main phases: innovation, productization, adoption & transition and service & variation. The innovation phase focuses on in-house development of tailor-made systems. The productization phase is where either spin-off companies take over development of existing systems, or software companies expand to new business with software products. In the third market phase, focus is on adoption and transition, where the number and variation of software companies and produces reduce due to market share battles. In the final phase before renewal, the focus in on competing with diversified services built on standardized processes, applications, and interfaces. (Tyrväinen et al., 2008)

For each of these phases, there are several different drivers of change, and software business opportunities, which give competitive advantage to companies, which can operate ahead of the market progression (Tyrväinen et al., 2008). Therefore, to be successful in software product management, the product manager must recognize the market phase the company and its products are.

From software product management point of view, especially during the productization phase, problems may arise when companies move from selling tailor-made software to standard product software. As the tailor-made software varies in many ways from standard product software, during the productization process, the maturity of the software product management should increase until all functions are in place (Artz et al., 2010). According to Harkonen, Haapasalo and Hanninen (2015) the characteristics of productization vary depending on what is being productized – products, services, software, or technology. Smaller companies appear more often during the productization phase, leveraging emerging standards ready to be exploited (Tyrväinen et al., 2008).

In such software start-ups, the focus early in the product development is to release the product as quickly as possible to verify the product/market fit, and then adjust the direction according to feedback. At this stage, start-ups often discard traditional software process activities, while taking advantage of a developmental and iterative prototyping approach, utilizing integrated tools, and outsourcing complex parts to third parties. The need to restructure the software product and processes, however, increases as the company grows. The start-up's initial gain of agility and speed is counterbalanced by more controlled processes as the start-up faces increase of customers, employees, and product features. The most significant challenge for early-stage software start-ups is

finding the balance between being fast enough to enter the market early and managing the accumulating technical debt. (Giardino et al., 2016)

In a survey from Product Focus (2019, pp. 11), the most important issue for product management role was that "product management responsibilities are not clearly defined and overlap with other roles". This problem of not understanding the role of software product manager and software product management, may lead to making wrong business decisions, which have negative impacts on the development of companies and products (Springer & Miler, 2018). The problems are partially explained due to varying and confusing job titles for people who factually deal with issues related to software product management (Springer & Miler, 2018). The role descriptions are highly environment dependent – in addition to the archetypical product manager presented in section 2.4.2, Springer and Miler (2018) have proposed eight different personas for software product managers in different business environments.

The impact of software market phases to the software products and size of the organization are situational factors, and the differences in role descriptions are results of such factors. Situational factors are described in more detail in chapter 4 as part of the process improvements related to software product management. The role of software product management, and the maturity of software product management processes, can be assessed and improved systematically by applying a product management framework. A selection of different Software Product Management frameworks is presented in the next section.

2.5 Product management frameworks

According to Fricker (2012) and Maglyas et al. (2017), there has been several different attempts to create a suitable framework for product management. Some product management frameworks describe the activities regardless of the business domain e.g. Pragmatic Framework (Pragmatic Institute, 2019), while some focus purely on software products (van de Weerd et al., 2006, Ebert, 2009, Kittlaus & Fricker, 2017). The frameworks have overlapping parts but are organized differently and use varying terminology to describe similar activities (Maglyas, 2013).

Fricker (2012), Maglyas (2013), and Maglyas et al. (2017) have made extensive analysis of the current state of software product management reference models. The illustration below (figure 5) contains Fricker's (2012) view on a generalized model for software product management, which have been constructed from several different models.

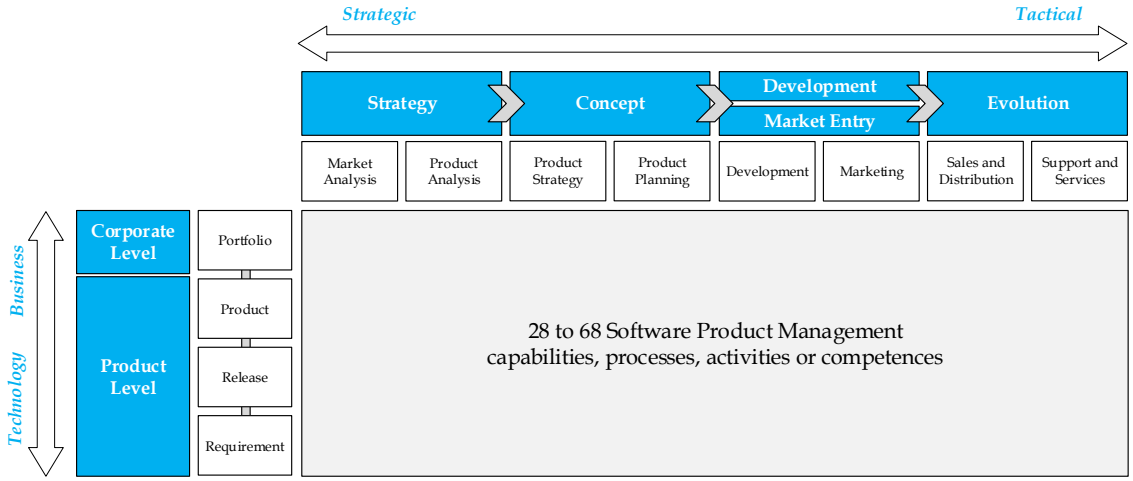

FIGURE 5 Overlay of software product management reference models (Adapted from Fricker, 2012, pp. 64)

The software product management frameworks or reference models are used for benchmarking companies, improving processes, and training software product management professionals. Depending on the framework, they refer to 28–68 capabilities, processes, activities, or competences that are typically structured according to product lifecycle phases, functional areas, interfaces to company functions, abstraction levels, or impact of decision making. (Fricker, 2012)

In Fricker's (2012) generalized overview, product artefact hierarchy (portfolio, product, release, requirement) is adapted from van de Weerd's (2006) reference framework. Lifefycle phases seen on the top are taken from Ebert's software product management framework (Ebert, 2007). The eight functional areas below the lifecycle phases and the split between corporate and product level are from software product management framework by Kittlaus and Clough (2009). Finally, the Pragmatic Framework (Pragmatic Institute, 2019a) has been used as basis to illustrate the focus shift from strategic to tactical on the top.

According to Ebert (2018), most software product managers have a technical career background and have grown into the discipline without any form of formal education. Such paths may produce one-sided product management approaches, while utilizing product management frameworks fulfils the demand better by providing a more holistic view. In his research conducted to worldwide business-to-business (B2B) software product companies, "lack of software product management discipline" was one of the three major challenges related to global software product management. Furthermore, he also found that there is high correlation of having a systematic competence training program, and product management success within companies. (Ebert, 2018)

The software product management practitioners (Ebert, 2018) and framework advocates (Kittlaus & Fricker, 2017, Blackblot, 2019f) all agree on the importance of having a structured approach. Kittlaus and Fricker (2017, pp. 257) summarize this well by stating "As long as product managers learn everything by doing, they frequently miss best practices already known in other companies". According to Blackblot (2019f), the product management process in an

organization depends on two key elements: professional training based on a "complete and structured product management methodology" and correct implementation of the chosen methodology in the organization. While external certification of product managers is rarely used as a hiring criterion in companies compared to for example project managers (Ebert, 2018), according to Kittlaus and Fricker (2017, pp. 257), "certification will certainly help in this process both to encourage personal development and to help companies to identify qualified software product managers".

Different software product management frameworks have been listed and compared to some degree by Fricker (2012) and Maglyas et al. (2017). "Product Management Trends and Benchmarks Report" published by Product Management Festival (2019) provides insight on which frameworks are being used by practitioners. The table (table 3) below summarizes how nine different frameworks have been present in the sources, and whether certifiable training is available for those frameworks.

TABLE 3 Different Product Management frameworks

#	Framework	Reference	(Fricker, 2012)	(Maglyas et al., 2017)	(Product Management Festival, 2019)	Certifiable?
1	SAFe	(Scaled Agile, 2019)			20,5%	Yes
2	Pragmatic Framework	(Pragmatic Institute, 2019a)	X	X	17,6%	Yes
3	ISPMA's SPM Framework	(Kittlaus & Fricker, 2017)	X	X	6,1%	Yes
4	Blackblot PMTK	(Steinhardt, 2017)			6,1%	Yes
5	AIPMM	(AIPMM, 2017)			6,0%	Yes
6	Reference SPM Framework	(van de Weerd et al., 2006)	X	X		No
7	SPM Framework	(Ebert, 2009)	X	X		No
8	SPM Competence Model	(Bekkers et. al, 2010)	X			No

It should be noted that the survey (Product Management Festival, 2019) has bundled the usage of these frameworks under one category "Methods/Processes in Use", and the listing included also other practices which do not include product management activities as has been defined earlier in this study. There were also some categories, where multiple practices were bundled into one – for example, a combination category "Scrum/Agile/Kanban" had 92,2% adoption rate, which shows that agile practices in one form or another

are widely used, but is not useful here in the comparison of product management frameworks.

According to recent study by Maglyas et al. (2017), the core software product management activities are vision creation (VC), product lifecycle management (PLM), roadmapping (RM), release planning (RP), and product requirements engineering (RE). Additionally, there are six supporting activities: strategic planning (SP), portfolio management (PM), product analysis (PA), product launches (PL), product support (PS), and product development (PD). The presence of those activities within the same nine frameworks as above, is presented and compared below in a table (table 4).

TABLE 4 Activities within Product Management frameworks (Adapted from Maglyas et al., 2017, pp. 37)

#	Framework	VC	PLM	RM	RP	RE	SP	PM	PA	PL	PS	PD
		Core activities					*Supporting activities*					
1	SAFe	X	X	X	X	X	X	X				X
2	Pragmatic Framework			X		X		X		X	X	
3	ISPMA's SPM Framework		X	X	X	X	X	X	X	X	X	X
4	Blackblot PMTK		X	X		X	X		X	X		
5	AIPMM		X	X	X	X		X		X	X	X
6	Reference SPM Framework		X	X	X	X		X		X		
7	SPM Framework			X	X	X	X			X		
8	SPM Competence Model		X	X	X	X		X	X	X		

The analysis (table 4) is based on previous work from Maglyas et al. (2017), who have mapped the activities against Pragmatic Framework (2), ISPMA's SPM Framework (3), Reference SPM Framework (6), and SPM Framework (7). The analysis of rest of the frameworks is based on short literature review on those frameworks in this thesis.

The five most relevant frameworks to study further are selected based on the following two criteria: 1) there must be certifiable training available for the framework, and 2) there must be reported real life usage for the framework. According to the established analysis (see tables 3 and 4), frameworks 1-5 fulfil both criteria, and the following subsections therefore go deeper into the following frameworks: SAFe, Pragmatic Marketing, ISPMA's SPM Framework, Blackblot PMTK, and AIPMM.

2.5.1 Scaled Agile Framework (SAFe)

Scaled Agile Framework (SAFe) is the most used framework for scale agile development across the enterprise (Product Management Festival, 2019). The first release of the framework was in 2011 (Leffingwell, 2011), and the latest version of the framework was released in January 2020 (Scaled Agile, 2020a). Over 500 000 practitioners have been trained on it, and over 280 partners provide training and consultation services for the framework worldwide (Scaled Agile, 2020a).

The SAFe framework is presented as the "Big Picture", which is available in four configurations suitable for different implementation scenarios (Scaled Agile, 2020c). The most comprehensive configuration, "Full SAFe" is illustrated in appendix 1. It is built around seven core competencies for a lean enterprise: lean-agile leadership, organisational agility, lean portfolio management, enterprise solution delivery, agile product delivery, team and technical ability, and continuous learning curve (Scaled Agile, 2020a). The lean-agile leadership competency is presented on the bottom as the foundation for the whole framework, while the other competencies shown on the left are used in different configurations of the framework (Scaled Agile, 2020c).

The competencies are organized into three layers of the framework: portfolio, large solution, and essential. The layers contain different activities and deliverables, which are executed by different roles and teams. The essential layer is present in all configurations. It is built on the practices within the lean-agile leadership, the team and technical agility, and the agile product delivery competencies, that are considered as core competencies for the framework. Finally, on the right side of the framework, there are additional roles and artefacts which may be optionally applied to different contexts. (Scaled Agile, 2020c)

In the model, there are activities and roles related to product management in the form of product management and solution management roles, located in the the essential and large solution layers. In addition, the role of product owner is crucial for the development activities within the essential layer. Many other aspects related to product management can also be found in various parts of the framework. (Scaled Agile, 2020c)

There has been much criticism about the presentation of agile principles in the framework (Turetken, Stojanov & Trienekens, 2017) and claims that it is not truly agile, as it has such heavily prescripted role-and-process design (Pancholi & Grover, 2014, Ebert & Paasivaara, 2017). Despite the criticism, industry reports on SAFe implementation (Turetken et. al, 2017) report many concrete benefits. Managers find the framework comfortable, because it has many ready-made role definitions, which have not existed in the agile approaches (Ebert & Paasivaara, 2017).

The training offering of Scaled Agile Framework has been highly commercialized by Scaled Agile. Ten core certifications and three advanced certifications are available for different roles and specific topics (Scaled Agile, 2020e). Scaled Agile also provides a roadmap describing the strategy and activities for implementing the framework (Scaled Agile, 2020d). In addition, one of the certi-

fications – the Certified SAFe Program Consultant – focuses on the implementation of the framework into an organisation (Scaled Agile, 2020f).

2.5.2 Pragmatic Framework

The Pragmatic Framework (Pragmatic Institute, 2019a) provides a "blueprint" of 37 product management activities, grouped into seven categories: market, focus, business, planning, programs, enablement, and support. These are shown in appendix 1. The activities above the categories focus to markets, and the activities below the categories focus to products (Pragmatic Institute, 2019b). Furthermore, the leftmost activities are more related to strategy, while the rightmost activities are more related to execution (Pragmatic Institute, 2019a).

Each box of the framework represents responsibilities and activities that must be performed by the product team. Each team requires individuals with suitable skills to be able to perform at a high level. (Pragmatic Institute, 2019c)

Pragmatic Marketing utilizes the framework to identify specific kinds of product managers and includes a triad of managers: a strategist, a technologist, and a marketer. The strategist, Director of Product Strategy, is responsible for strategic business-oriented practices. The technologist, the Product Manager, is responsible for strategic technology-oriented practices. And finally, the marketer, the Product Marketing Manager, is responsible for tactical market-oriented practices. (Fricker, 2012).

Pragmatic Institute was founded as Pragmatic Marketing in 1993, and they have taught more than 150 000 product management and marketing professionals at 8 000 companies in 25 countries. Current course offering contains product courses starting with Foundations, which is prerequisite for other courses. Focus, Build, Market, Launch, and Price courses go deeper into the different areas of the framework. All courses contain the possibility to become Pragmatic Management Certified, which exists in seven different levels. (Pragmatic Institute, 2019d, Pragmatic Institute, 2019e, Pragmatic Institute, 2019f)

Overall, the Pragmatic Framework and its supporting materials contain a general overview of the product management activities which is independent of business domain or industry (Pragmatic Institute, 2019). The framework has become one of the most well-known and popular frameworks due to active marketing and trainings conducted by Pragmatic Insight (Maglyas, 2013).

2.5.3 Blackblot Product Manager's Toolkit (PMTK)

Blackblot Product Manager's Toolkit™ (PMTK) is a comprehensive set of tools and supporting methodology, laying out best practices and processes, which assist in creating successful market-driven products. The toolkit is a systematic, managerial approach for planning and marketing of new products, and managing the life-cycle of existing products. (Steinhardt, 2017)

The two main processes covered by the PMTK are product planning and product marketing (Steinhardt, 2017). It also includes guidance to process effi-

ciency and other areas, such as legal documents and personnel management (Blackblot, 2019f). Appendix 1 shows the PMTK Action Model, which includes the main processes, their sequences, and activities within the model.

The Action Model is the main framework for the Blackblot PMTK, which also includes several other parts. The other two essential models in the framework are the Flow model, which applies the Action Model in a stage-gate process, and the Task Model, which synchronizes the PMTK templates with the concepts of teams: product management team and product definition team. Support and Concept Models include mostly supporting illustrations of different responsibilities and makeup of teams and process bits. (Blackblot, 2019a)

PMTK was originally written by Blackblot as internal document templates, so that product managers could focus more on the content instead on the methods. The templates are stated to be intentionally broad and general in nature due to differences in each business. The users of the methodology will need to modify the templates to fit their specific needs. (Blackblot, 2019b)

The toolkit is used by over 5 500 product managers and marketers worldwide, and it is also offered as a Blackblot Strategic Product Manager (SPMR) training leading to an accreditation called "Blackblot Product Management Professional (BPMP)". Over 3 000 professionals have been trained in the methodology since 2004. It is possible to also have training on advanced topics, such as leading product organisations, product marketing, market-value pricing, and product management for service providers and manufacturers. (Blackblot, 2019c, Blackblot, 2019d, Blackblot, 2019e)

Blackblot also provides a process for introducing and implementing the Blackblot PMTK Methodology within a company. The phased process, including the customization of the methodology, is guided by the documents and tools learned in the training programs and the PMTK Implementation Plan. (Blackblot, 2019f)

2.5.4 ISPMA's SPM Framework

SPMBoK (SPM Body of Knowledge, 2019, Kittlaus & Fricker, 2017) is one of the first product management approaches, that specializes to software products and integrates together various areas of product management. The ISPMA's SPM Framework is part of this approach and it has been developed since 2009 by ISPMA (International Software Product Management Association), based on best practices from the enterprises and academic research. It has been developed based on software management framework models by van de Weerd et al. (2006), Ebert (2007), and Kittlaus and Clough (2009).

The framework defines the major functions involved in software product management ("Core SPM") alongside with tasks to be involved with via "Participation" or "Orchestration". The Core SPM consists of Product Strategy and Product Planning as the main functions which unite the core software product management activities. The horizontal structure (columns) in the framework follow the functional areas to be implemented in a product or company level for

a software organisation. Vertically within each column, the structure is based on top-down approach, i.e. the most strategic and long-term activities are on the top, and downwards the focus changes to more operational and short-term. The framework is further detailed in chapter 3. (Kittlaus & Fricker, 2017)

Certification programs to become a "ISPMA Certified Software Product Manager" are offered in two levels: Foundation and Excellence level. The Foundation level teaches the full spectrum in the framework, while the Excellence level goes deeper into each aspect. Excellence level courses are available for the Core SPM as separate courses (Product Strategy and Product Planning), but also for the Participation and Orchestration. (ISPMA, 2019a, ISPMA, 2019b).

Foundation level courses are targeted to participants with up to 5 years of practical experience in software business, while the Excellence levels are targeted to product managers with either Foundation level certificate or comparable experience of at least 3 years. (Kittlaus & Fricker, 2017)

Trainings are offered by independent training companies, and the certification exams are administrated by independent certification authorities (ISPMA, 2019a). There are over 800 certified members worldwide (Kittlaus & Fricker, 2017), of which of roughly 600 are in the public listing (ISPMA, 2019c).

2.5.5 AIPMM

The Association of International Product Marketing and Management (AIPMM) promotes worldwide excellence in product management education. They do it by maintaining Product Marketing and Management Body of Knowledge (ProdBOK). AIPMM has been established in 1998. (AIPMM, 2020a)

The ProdBOK consists of "Seven Phase Product Life Cycle" and the standardized best practices and certification around it (Geracie & Eppinger, 2013). The ProdBOK is commercially available as a book (AIPMM, 2020c). The framework is illustrated in appendix 1.

The seven product lifecycle phases in the framework are: conceive, plan, develop, qualify, launch, deliver, and retire. First four phases are related to new product development or acquisition, while the last three phases are related to commercialization and manufacturing operations. Each product can be seen to be in the intersection of five knowledge areas: market, customer, business, organisation, and product. (Geracie & Eppinger, 2013)

Each lifecycle phase contains activities and deliverables. Activities and deliverables with the lifecycle phases are grouped into different knowledge areas: organisation, business, customer, product, and market. There is a "gate" between each phase where decisions to proceed (or not to proceed) to the next phase is taken. (AIPMM, 2017)

Since 2004, it has been possible to become Certified Product Manager, Certified Product Marketing Manager, or other product related expert via their certification program (AIPMM, 2020a). The trainings are available via various partners, the main partner being 28OGroup (AIPMM, 2020b).

2.6 Summary

Products are combinations of goods and services, which vendors combine in support of their commercial interests to transfer rights to customers. Products are created to deliver core customer value via the actual product, which is usually augmented by additional services and benefits. Product management governs a product and processes related to it from its inception to its withdrawal from the market. Product manager is the person responsible for the product success – or in many cases, multiple products within a larger portfolio.

Software products are products whose primary component is software. First software products were conceived in the 1960s after the separation of software and hardware. By the 2020s, the variety of software products covers a huge range of different offerings to both business-to-business and business-to-consumer segments. The software products' special characteristics compared to physical goods include marginal manufacturing and distribution costs, easy changeability and updating over time, high release frequency, high complexity, and increasing returns through network effects.

Succeeding in a fast-moving software industry depends on the business model. In the software industry, there are many alternative and complementary business models to combine with the product and service offering. Software product management as an established discipline is much younger than the software products, and for decades the software product managers have been learning by doing in the lack of education and best practices. Software product management core activities are vision creation, product lifecycle management, roadmapping, release planning, and product requirements engineering.

The software product manager is seldom responsible of all activities of the intangible software product, but is usually accompanied by product marketing manager, project manager or the agile role of product owner. The software product manager works increasingly with distributed agile teams and embraces the global software development paradigm to stay on top of competition in the market. Software product management is therefore not without challenges – the high-tech markets move fast, start-up businesses rise and fall, new paradigms and technologies change the playing field, and the products are increasingly more complex, while expected to deliver increasing value over the product lifetime.

Fortunately, there are frameworks, which can assist organisations in implementing software product management principles and practices. For many product management frameworks, there are commercial training and certification offerings available to facilitate learning. In addition to utilizing generic product management frameworks, there have been efforts to create frameworks specialized for software product management. The approaches and scope in the frameworks differ. Pragmatic Framework can be applied regardless of the type of business – software or not – however there is heavy focus in the framework on the marketing aspects. Blackblot Product Manager's Toolkit focuses on a

template-driven approach, intending to provide a good structure for tailoring your own process focused on content. The AIPMM model is built around the product life cycle model of the products and a stage-gate model within that life cycle. SAFe focuses on scaling agile development across the enterprise and considers software product management activities as part of the very detailed and arguably complex framework. The ISPMA's SPM Framework specializes purely on software product management and seems to be matching best the identified activities for software product management. While lacking "ready-made" processes as some of the frameworks do, it has a commercial training offering available and a solid background from both industry and academia. The ISPMA's SPM Framework will be utilized in this thesis for further research, and it is the subject of the next chapter.

3 ISPMA'S SPM FRAMEWORK

In this chapter, we investigate more closely the ISPMA's SPM Framework. It is the only framework covered in the previous chapters, which specializes to Software Product Management. According to Fricker (2012), it is unique also the way that the contents and scope of the framework continue to develop by taking in the latest understanding of the discipline by active discussion between practitioners in the knowledge network.

The ISPMA's SPM Framework structures software product management practices along functional areas. In the figure below (figure 6), the horizontal structure (columns) follow the functional areas to be implemented in a product or company level for a software organisation.

PARTICIPATION	CORE SPM		ORCHESTRATION			
STRATEGIC MANAGEMENT	PRODUCT STRATEGY	PRODUCT PLANNING	PRODUCT DEVELOPMENT	PRODUCT MARKETING	SALES AND DISTRIBUTION	SERVICE AND SUPPORT
Corporate strategy	Positioning and product definition	Product life-cycle management	Engineering management	Marketing planning	Sales planning	Service planning and preparation
Portfolio management	Delivery model and service strategy	Roadmapping	Project management	Customer analysis	Channel preparation	Service provisioning
Innovation management	Sourcing	Release planning	Project requirements engineering	Opportunity management	Customer relationship management	Technical support
Resource management	Business case and costing	Product requirements engineering	User experience design	Marketing mix optimization	Operational sales	Marketing support
Market analysis	Pricing		Quality management	Product launches	Operational distribution	Sales sup-port
Product analysis	Ecosystem management			Operational marketing		
	Legal and IPR management					
	Performance and risk management					

FIGURE 6 ISPMA's SPM Framework (SPM Body of Knowledge, 2020)

Vertically within each column, the structure is based on top-down approach, i.e. the most strategic and long-term activities are on the top, and downward the

focus changes to more operational and short-term. However, the detailed dependencies of the activities within and between each column are more complex than can be fully illustrated in such matrix structure. (Kittlaus & Fricker, 2017).

In the structure, there is an additional grouping of columns to "Core SPM", "Participation" and "Orchestration". It indicates that the software product manager participates in Strategic Management, is directly responsible for Product Strategy and Planning, and orchestrates Development, Marketing, Sales and Distribution, and Service and Support. In large companies, corporate level functions are typically responsible for market and product analysis activities, collaborating with participating product managers. However, in small companies, product managers can be responsible of those activities alone. This is highlighted in the participation column, as knowledge elicitation related to market and products is part of the core SPM responsibilities. (Kittlaus & Fricker, 2017).

The background organisation behind the ISPMA's SPM Framework is International Software Product Management Association (ISPMA), which was established in 2009. It is a non-profit organisation whose fellow members are software product management experts from the industry and academia. The goal of ISPMA is to foster software product management excellence across industries. (Kittlaus & Fricker, 2017)

The syllabi included to the certification programs related to the framework are freely available online (ISPMA, 2019b). The curriculum and Certifiable Body of Knowledge (SPMBoK) has also become a basis for variety of commercial training offerings and university courses (Kittlaus & Fricker, 2017).

Foundation Level certification and syllabi provides an overview of all elements of software product management and covers both core, participation, and orchestration areas of the ISPMA's SPM framework (ISPMA, 2016a). Excellence Level syllabi and certifications, which are structured according to the framework verticals go deeper into the different areas of the framework. Excellence levels Product Strategy (ISPMA, 2016b) and Product Planning (ISPMA, 2016c) focus on the "Core SPM" activities of the framework. Respectively, Strategic Management (ISPMA, 2016d) focuses on the "Participation" column and Orchestration (ISPMA, 2017) to the "Orchestration" areas of the framework. While the foundation level certification focuses on knowing and understanding the framework (ISPMA, 2016a), the aim of the excellence levels is also to train practitioners in applying the framework in practice via practical exercises (ISPMA, 2016a, ISPMA, 2016b, ISPMA, 2016c, ISPMA, 2016d, ISPMA, 2017).

The most comprehensive written guide to the ISPMA's SPM Framework is a book "Software Product Management – The ISPMA-Compliant Study Guide and Handbook" by Hans-Bernd Kittlaus and Samuel A. Fricker (Kittlaus & Fricker, 2017). Software product management is a continuous activity over the non-predetermined lifecycle of the software product. Since the software product management consists of a plethora of separate tasks, it does not make sense to view as a one consequent process (Kittlaus & Fricker, 2017). The following sections describe further the different areas of the framework.

3.1 Core SPM

Product Strategy and Product Planning are the core areas of the ISPMA's SPM Framework. Together they answer to the question what the software product will be, when and how it will be developed, and how it will be used by the customers. The product manager defines the software product, sets targets and secures commitment from stakeholders for the evolution of the product over its life cycle. The software product manager is responsible for the decisions taken and takes leadership in managing the different aspects of the product strategy and planning. The following sections explain in more detail the activities within product strategy and product planning.

3.1.1 Product Strategy

Product Positioning and Definition means activities which surround the software product's vision and its intended business model. The product managers need to have a clear definition about the customer segments they are targeting, in order to develop a deep understanding of customer needs. This understanding is then translated into value propositions, which should resonate with the defined customer segments, and help in selling the products and services. Especially in Business-to-Business software market, the products often require additional offerings to be combined into a holistic solution, or a "whole product" with partner offerings. A critical factor on product positioning and definition is also to consider the competition and identifying competitive advantage for the product. Competitors include direct competitors having similar products, indirect competitors selling different products satisfying the same needs, and other alternatives the customers may have – for example solving the problem on their own. The identified competitive advantages form a basis for various competitive strategies for the product: low cost strategy (cost leader), differentiation strategy (uniqueness) or focus strategy (niche leader). When thorough understanding of customers and attractive value propositions are combined with an understanding of competitive offerings, the product can be positioned clearly and positively in the market and target customer's mind. (ISPMA, 2016b)

The product definition describes the product on an abstract level – what is it and what is it not – with the target to give guidance to the direction and decisions around the software product. The definition includes users, intended use, functionality, quality, user experience, technical constraints, compatibility, customization, delivery model, and finally, the whole product offering. The product definition activities involve many stakeholders such product management, marketing, sales, and development in an iterative process, which evolves over the product's life cycle phases. The resulting definition of the product is an important element of the product positioning and scoping the product. (Kittlaus & Fricker, 2017).

As part of the product positioning and definition, the software product manager must also select and manage the sales channels related to the delivery of the product to the customer. Sales can be made via either physical or virtual channels, which each have different characteristics that the software product manager must comprehend. Physical channels include sales performed by humans, either by the company's own sales force or by various partners. Such partners can be for example original equipment manufacturers, service providers, value added resellers, independent software vendors, and system integrators. Virtual channels such as internet sales, is a major channel to considered for the product. The product manager must select the channels appropriately for the product and market, and typically the result is a mix of different channels which also means management need for channel conflicts. (ISPMA, 2016b)

Delivery Model and Service Strategy includes activities around how the product is made available to the customer, and which kind of human services are included in the product. Two main variants of the delivery model for software products the software product manager must consider are on-premise and SaaS (Software as a Service). In an on-premise model, the customer is sold a license for the software product and they run it in their own operating environment. In SaaS, the vendor of the software product is responsible for the operating environment based on a service contract with the customer, which may include different degrees of operations and maintenance. (ISPMA, 2016b)

The software product manager is responsible of the whole product offering, which can include various types of services, that are provisioned either internally or via partners. Product-related human service can include training, installation, customization, operations, and maintenance. Additionally, product-related services may include custom software development or system integration. The product-related services are crucial for the success of the software product, and therefore belong on the software product manager's table concerning product strategy. (Kittlaus & Fricker, 2017)

Sourcing covers management of resources required to implement the product strategy. It is required in many aspects to fulfil the whole product offering with compelling products and services. For software products, the most important resources are skilled humans, which can be either employees, or hired from outside. In addition to partnering with external companies for software product development efforts, other software vendors may also provide ready-made components, platforms, plugins, infrastructure, or data. The make-or-buy decisions done in with stakeholders such as development, marketing, and service organisations can have major impact on the time-to-market, quality, or financial performance of the software product. (Kittlaus & Fricker, 2017)

Business Case and Costing activities include building business cases, costing, and financial models. A financial business case is a decision support approach, where investments and benefits are quantified and compared, so that investments can be justified, and profitability can be tracked over time. Key inputs for a financial business case are revenue model and a cost model, which both must be understood by the software product manager. Software product managers

can choose from a wide selection of business models to create a revenue model for a software product. A business may combine multiple revenue models, and each revenue model can rely on multiple revenue streams, which is typical for software industry. As a simple example, a classic license model usually combines license revenue stream with support and maintenance revenue. On the cost side, for software products the cost of revenue is highly variable. In addition, research and development, sales and marketing, administration, and other operating expenses are typically also included in the business case. Because many software products have a low marginal cost, the financial analysis of the business case often focuses mainly on revenue: costs are treated as unchangeable and the main question is that can a suitable price structure be found which shows that profit could be made? Software product managers typically build financial models to simulate different scenarios based on "what-if" questions and how profitability can be studied under various conditions. (ISPMA, 2016b)

Pricing means the activities which are required to set, communicate, and negotiate software prices in a credible way. The pricing strategy starts from understanding the customer segments and value delivered by the product to the customers. The product manager must ensure that the price and value can be communicated, pricing policies are accepted, and the price levels are set to maximise the profitability of the product. The product manager utilizes different pricing models depending of the life cycle stage of the product, market penetration strategy, delivery model and distribution channels. For software products, value-based pricing approach is preferred, instead of cost-based pricing typical to human services in the software business such as software product development projects. Pricing is a continuous source for conflict within an organisation, and the product manager must keep in mind that the price does not sell the product: the product must first fit the customer need. As the pricing has a major impact on the product's economic success, the pricing is a balance of finding out what is suitable for the market and what optimizes the long-term financial benefits of the software product vendor. (Kittlaus & Fricker, 2017)

Ecosystem Management means activities related to the role and strategy of the software company and its products in a software ecosystem. A software ecosystem can be defined as a "set of businesses functioning as a unit and interacting with a shared market for software and services, together with relationships among them" (ISPMA, 2016b, pp. 15). The software product vendor must establish a role in the ecosystem, which directly influences different factors of the product strategy such as the positioning, pricing, and dependencies from other ecosystem players to own product planning. Three major ecosystem strategical roles are keystone, dominator, and niche player. Most companies within a software ecosystem follow a niche player strategy and focus their business on narrow area of expertise where they can do profitable business. Keystone players provide the core of the innovation in the ecosystem, and behave in favour of other ecosystem players, gaining benefits of the overall ecosystem success. The dominators are the opposite of the keystone players, attempting to exploit a large portion of the ecosystem by occupying or even monopolizing a growing

number of niches. The ecosystem management includes the wide variety of management of relationships and partnerships between different ecosystem players, such as competitors, acquisition targets, channel partners, other vendors, influencers, and customers. (ISPMA, 2016b)

Legal and Intellectual Property Rights Management covers activities related to legal aspects to software products. Software product managers need to have an overview of the legal framework and risks surrounding the product, although details are typically handled by legal experts. Four major areas to manage are contracts, protection of intellectual property, open source, and data protection. Contracts include the legal aspects relating to how the software is acquired by the customer: scope of license or service, type of charges, warranty, liability, maintenance, and so on. Protection of intellectual property is critical since the development of the software product requires significant investments and copies of the software can be easily done. Trademarks, trade secrets, copyrights and patents form the legal basis for the protection of intellectual property. Open source is increasingly important part of the software products, as it has a great potential to reduce development investments. However, one must manage the copyright protection and licensing aspects of both owned and third-party open source components, as with any other software, to ensure a solid legal basis for the software business. Lastly, the software product manager must ensure data protection laws are followed, which have significantly different effects depending of the delivery model of the product. In addition to these four areas, the product managers also often need to take into consideration other aspects such as governance, finance, delivery commitments, supply-chain, product liability, or blacklisting of software components in certain countries. (ISPMA, 2016b)

Performance and Risk Management means activities around continuous tracking and analysis of the business performance, and risks related to the software product. Business performance is usually followed via various key performance indicators (KPIs), which ideally addresses all elements of the product strategy. Performance can be measured for example from customer, financial, internal business, learning, and innovation perspectives. The choice of right measures that support decision making is critical, as so-called "vanity metrics" may make the company "feel good", but do not provide actionable information. Usage of a software product can be measured unlike other products: for example, in web-based environments users can be tracked to develop metrics on the usage of features within a product, which enables data-driven decision making. The ability to manage risks effectively is critical for successful software products. The risks can be grouped into three major categories: product risks, customer risks, and market risks. Product risks are related to getting the product right, while customer risks are related to building the right product fitting the customer's problems. Finally, market risks are related to building a profitable business. The risks differ depending of the product's life cycle and the mitigation strategies must be applied against the most relevant risks of the time. (Kittlaus & Fricker, 2017)

Kittlaus and Fricker (2017) suggest that the activities related to product strategy should be implemented in such a way, that different activities have a predefined cadence. Periodic activities should be done regularly, monthly or less frequently, while triggered activities should be done either also, or solely when a specific event or request happens. The classification of activity type per framework area are described in the table below (table 5).

TABLE 5 Cadence of product strategy activities (Adapted from Kittlaus & Fricker, 2017, pp. 117)

Attribute	Periodic	Triggered
Positioning and Product Definition	X	X (if new)
Delivery model and Service Strategy	X	X (if new)
Sourcing	X	X (if driven by current situation)
Business Case and Costing	X	
Pricing	X	X (if driven by current sales situation)
Ecosystem Management	X	X (if driven by ecosystem members)
Legal and IPR Management	X	X (if driven by Sales or Development)
Performance and Risk Management	X	

3.1.2 Product Planning

Product Lifecycle Management considers activities related to product planning options during its lifecycle, both from the product's unique point of view, and market point of view. A software product evolves through series of phases over its own lifetime: conception and creation, market introduction, growth, maturity, decline, and withdrawal. A product evolves through these phases and the product managers must have good understanding of the characteristics of the phases. Challenging decisions must be done in phase changes, and they often create resistance within an organisation, since the way of working needs to change as well. The markets where the product is being offered, has life cycles as well. The market – or the category of products where the managed software product belongs to – moves in its own phases independently of the product. Product category phases are technology adoption, growth market, mature market, declining market, and end of life. A software product manager must understand to which category the product belongs, and in which lifecycle phase this category currently is, so optimal product planning can be done. The options differ, depending on the market phase, and if product planning is done for a new product entering the market in a specific phase, or if the product is already established in the market. (ISPMA, 2016c)

Roadmapping means activities related how the product vision is developed over time, according to the product strategy. The roadmap is one of the main tools for a software product manager for discussing and communicating product vision and strategy with different stakeholders both internally and externally. Internal roadmaps are at a suitable level of detail so that collaborating corporate functions can understand their contribution towards product success, which is important for the internal commitment from other corporate functions.

Roadmaps are planned in the strategic timeframe, which usually is between one to five years, depending of the product and the company. The software product manager must consider market opportunities, threats, improvements, cost reductions, revenue generation, and change of regulations when planning the content and order of the items in the roadmap. The concrete roadmaps are typically illustrated as various graphical representations, and there may exist different versions for different uses. The degree of detail and reliability of the roadmap is typically more detailed in short-term, than in the long-term. The roadmap sets expectations for release dates for high-level items, which usually are set to correlate with important market events, such as product launches or conferences. Roadmaps are also important for communicating to external stakeholders the direction of the product. External roadmaps create reactions, which validate the roadmap content, but also are used to build market trust towards continuous long-term investment to the product. (ISPMA, 2016c)

Release Planning covers activities related to defining content and schedules for one or more product releases, which are aligned with the product roadmap and the software organisation's capability to deliver. The release planning process varies depending on the product's life cycle phase. In early phases the release planning has higher degree of uncertainty, than in a more mature software product. Typically, there is not enough resources to implement all requirements for a product, and prioritization is required. The requirement prioritization is central for release planning, as it focuses the discussion on what is the most important functionality to be implemented and delivered timely to potential customers. Important inputs for the iterative release planning process are also understanding of the product itself, organisational environment, parameters of development, and the product life cycle dependencies. Parameters of development are various: stakeholders, requirements, preferences, and many constraints such as cost. In the release planning process, release plan alternatives are created, evaluated, and negotiated before decisions are made on the release plan. Results from release planning are typically documented in a release plan, that all internal stakeholders agree to. The release plan for software products consists of at least planned release dates, resource estimations, and the selected requirements to implement. The type of the software product affects the scope of the release planning. For example, for software products which are in a vendor-controlled environment, release planning could be considered necessary only for larger changes, and smaller changes could be more flexibly implemented, delivered, and experimented on. However, in customer-controlled environment, the planning may need to be more rigid. (ISPMA, 2016c)

Product Requirements Engineering means activities related to elicitation, analysis, selection, and validation of requirements in the product and market context. The product requirements are not limited only to the software product itself, but also to other elements of the whole product, including for example product-related services or legal aspects. Requirements come from variety of sources, and not only from customers, as the product organisation needs to bring innovative ideas into the software product. The number of requirements

can be large, and typically good tool support is required to manage the process. Compared to classical project requirements engineering for tailor-made software focusing on individual customer cases, the product requirements engineering focuses on aspects which provide value for a larger group of existing or potential customers, and product-related stakeholders. The success of the product is measured by ability to meet the market needs and requirements better than the competition, instead of contract fulfilment. (ISPMA, 2016c)

Kittlaus and Fricker (2017) suggest that the product planning activities should be implemented so, that different activities have a predefined cadence. The classification of activity type per framework area is described below (table 6).

TABLE 6 Cadence of product planning activities (Adapted from Kittlaus & Fricker, 2017, pp. 182)

Attribute	Continuous	Periodic	Triggered
Product Life Cycle Management		X	X (if new)
Roadmapping		X	X (if changes need to be reflected and a presentation is required)
Release Planning		X	X (if changes need to be reflected)
Product Requirements Engineering	X	X	

Continuous activities should be done more often than once a month, and periodic activities should be done regularly in a monthly basis or less frequently. Finally, the triggered activities should be done either also or solely when a specific event or request happens.

3.2 Participation

The Participation area of the ISPMA's SPM Framework includes functional area of Strategic Management. The product manager participates to the activities in this area, but the responsibility and leadership of the activities belong to the executive management of the company. To put it shortly, the executive management sets goals and constraints for the company's product portfolio, while the product manager influences the company's strategy. This area can be considered to represent the interface between software product management, and the company's executive management. This section describes shortly different areas of strategic management in the framework.

Corporate Strategy covers activities around corporate strategy, and its relation to the software products. For the software product manager, it is important to know which strategic principles and approaches are utilized in the organisation, in order to collaborate with executive management. A product manager with good understanding of the corporate strategy can both provide useful in-

put into company level strategy processes, and apply the guidance given from executive management for the product appropriately. (ISPMA, 2016d)

Portfolio Management focuses on activities around the whole portfolio of products in an organisation. The target of the portfolio management is to ensure having the right product mix for a given company strategy. Strategic decision making on what to further invest in, is done across the whole product portfolio. In portfolio management, optimization is done between different products in the portfolio, while the product manager's focus is on a single product and its lifecycle. Therefore, the product manager needs to know the importance and relevance of an individual software product in the portfolio and manage it accordingly. The software product manager provides inputs for the portfolio management process and follows its decisions subsequently. (ISPMA, 2016d)

Innovation Management contains activities related to having an environment suitable for innovation, how the right ideas are selected, and how ideas can be developed further. Software product managers often focus to product innovations related to features and quality aspects, which bring more value to the customer. The innovations can also be made on how to market the products, improve processes, or change the business models. Formalizing a process for innovation is extremely difficult, and it is often more productive to ensure that the software organisation's environment supports innovation. A great initial idea requires iterations to mature into something with executable potential. The iteration process, which is led by software product managers, involves different stakeholders, and may include prototyping and concepting with customers to ensure the value and optimal user experience. (ISPMA, 2016d)

Resource Management includes activities which ensure that the necessary resources are available, so that the company can achieve its business goals according to the corporate strategy. The software product manager may have direct access to product budgets and resources to ensure the product strategy can be executed, or the resources can be under executive management or other organisational control. In any case, the product manager should have a good overview of the key resources required to deliver the value propositions according to the product strategy. (ISPMA, 2016d)

Market Analysis covers activities in analysing the characteristic of both existing and future markets related to the product. Defining what a market is, is central for market analysis. One way is to split it into a generic market, relevant market, and the product market. The markets are influenced by different market forces, industry forces, and trends, that are evaluated during the market analysis. The results of the market analysis are utilized by the software product manager in number of different activities, such as product positioning and roadmapping to ensure that product fit-to-market is optimal. (ISPMA, 2016d)

Product Analysis contains activities which aim to achieve a comprehensive view of the current and previous state of the business, and technical performance of the software product. The result of the product analysis gives an overview of internal and external product status, containing facts such as number of customers, cost, and revenue. The focus of the product analysis depends

on the life cycle state of the product. Different key performance indicators (KPIs) related to finance, customers, development, and product-usage are useful of monitoring the product performance. Based on the product analysis, the software product manager can position the product compared to the targets and correct the direction accordingly in related activities such as performance management, life cycle management, and roadmapping. (ISPMA, 2016d)

3.3 Orchestration

Orchestration area in the ISPMA's SPM Framework contains four functional areas from which the software product manager depends on in realizing the product vision and executing the product strategy. The software product manager could be described as a "internal customer" which orchestrates the different company functions of development, marketing, sales and distribution, and service and support, but delegates responsibility. Development function implements the software, while marketing identifies and wins the customers. Sales and distribution close the deals, nurture the customer relationship, and generates revenue. Service and support facilitate the product use over its lifetime. Each of the orchestrated functions have their own responsibilities, tasks, and objectives, which can be a source for conflict for software product management in a cross-functional product team. The software product managers need to maintain a suitable mindset for navigating the organisational landscape, so they can orchestrate and achieve influence without authority. This section describes shortly these different areas of orchestration in the framework.

3.3.1 Development

The development function has a large impact on the product's functional and non-functional capabilities and qualities, including the user experience. Successful collaboration with development is therefore crucial for software product managers to achieve product success. (ISPMA, 2017)

Engineering Management includes all activities which are related to the software development projects. It includes product architecture, development processes and tools, knowledge management, configuration management, resources and skills, sourcing of development, and estimations. (ISPMA, 2016a)

Project Management contains activities related to execution of a release plan in software development, which are typically done in a project organisation. Depending of the chosen development methodology, the cooperation model between development and software product management differs. The project is usually responsible at least for internal documentation but is also an important contributor to the external product documentation. (ISPMA, 2016a)

Project Requirements Engineering covers activities which are within the development project responsibility and follow a similar process than the product

requirements engineering process. The product requirements are converted into project requirements and refined further as implementable features. Project requirements also take care of internal needs related to the development, as they are found during the development project. (ISPMA, 2016a)

User Experience Design contains activities related to the users' interactions with the software product, and how to affect the user's behaviours, attitudes and emotions concerning the product. The user experience design process considers human-system interactions, user interfaces, ergonomics, and the context of the product use. The user expectations can vary much depending of the different software and market segments, which needs to be considered already in the product strategy. (ISPMA, 2016a)

Quality Management includes activities related to the technical quality of the software product. Such activities are test strategy, infrastructure, support concept, issue database, quality assurance and quality forecasting. (ISPMA, 2016a)

There are areas of potential conflict, and the software product managers need to work continuously and pro-actively with the development, in effort to prevent them. Typical areas for conflict with development include release planning, project requirements engineering, and user experience design. The software product managers need to adopt their collaboration methods to fit the development practices used by the development function. The overall focus for the software product management considering development should be in the following areas: release scope and timing, acceptance of results, execution of plans, synchronization of product versus project requirements, estimates, and resource management. (ISPMA, 2017)

3.3.2 Marketing

The marketing function of a software organisation is responsible for areas related to preparation and support of product sales. Good collaboration between the software product manager and marketing is vital in creating a successful product. While the product manager focuses on what to build, the marketing focuses on how to communicate the product to the target markets. (ISPMA, 2017)

Marketing Planning contains the development and agreeing on plans and budgets for marketing activities within a given timeframe. The marketing plans need to be synchronized with corporate strategy, product strategies, and sales plans. A marketing plan can be product specific, or for groups of products. (ISPMA, 2016a)

Customer Analysis focuses on the analysis of existing and potential customers, in relation to finding additional business opportunities and retaining existing customers. The analysis is done frequently to customer or groups of customers. (ISPMA, 2016a)

Opportunity Management is a continuation of the customer analysis, since it contains the activities related to pursuing the identified business opportunities. The opportunity management includes for example formulation of new product

requirements, developing and implementing new marketing tactics, and close cooperation with sales. (ISPMA, 2016a)

Marketing Mix Optimization is the selection, execution, and coordination of marketing channels and partners for the product. The product strategy defines the product ecosystem, which can be utilized for marketing appropriately for the marketed product. (ISPMA, 2016a)

Product Launch contain activities related to the introduction of new product, version, or features to the markets. Marketing orchestrates activities internally, and with external partners to ensure attention of the existing or potential customers related to the product launch. (ISPMA, 2016a)

Operational Marketing contains the actual activities related to the execution of the marketing plan. The operational marketing is tracked with relevant metrics, and corrective actions are taken when necessary. (ISPMA, 2016a)

With marketing, there are typical areas of conflict to be understood by the software product manager. Such areas can be either on strategical, product, or communication level. On a strategical level, conflicts can often arise from whether to invest on brand or product marketing. On product level, roadmap and requirements are constant source for conflict also with marketing. A major communication related conflict area is product launches. Launches of new products, versions, or features can be hectic, and cause conflicts, as they are typically time critical. The software product manager should focus on the following areas when orchestrating marketing activities: positioning of product in a marketing plan, plan execution, product launches, channel and partner management, participation to marketing events, and balance between brand and product marketing. (ISPMA, 2017)

3.3.3 Sales and Distribution

The sales organization is responsible for all sales activities of a company. Sales is made by selling products either directly to customers, or through channel partners. Collaboration with sales organisation and its processes by influencing sales motivation and enabling sales is important for the software product manager to achieve product success. Once the sales are made, the product needs to be distributed to the customer. Distribution can reside in the sales organisation or be within some other delivery organisation. The type of distribution differs depending of the whole product aspects of the software product, which the product manager must orchestrate. (ISPMA, 2017)

Sales Planning contains activities related to development and agreeing of plans concerning sales activities within a given timeframe. The sales plans need to be synchronized with corporate strategy, product strategies and marketing plans. The sales plan can be product specific, or for a group of products. (ISPMA, 2016a)

Channel Preparation covers the activities related to preparing the channels in time for selling a new product, version, or feature. It includes for example

sales skills management, websites, and customer reference stories. (ISPMA, 2016a)

Customer Relationship Management is the coordination of the company's interactions with current and potential customers and clients. It includes the customer communication, knowledge management, and management of the customer requirements. The focus must be both on short-term sales success and nurturing long-term customer relationships. (ISPMA, 2016a)

Operational Sales consists of the actual execution of the sales plan and tracking its success. When necessary, corrective measurements are taken if the metrics deviate from the plan. The operational sales also include offers, negotiation of contracts, and management of both. (ISPMA, 2016a)

Operational Distribution includes activities related to ensuring smooth order and distribution process related to the software product. The process needs to be stable and lead to correct billing of the customers. In case the software product needs physical distribution, sufficient supply must be also ensured. (ISPMA, 2016a)

Typical sources for conflict with sales for the software product manager are related to getting product feedback from sales, sales incentives, price and discounting, short-term customer requirements versus long-term market requirements, and impact on software product management engagement with customers. In the area of distribution, conflicts may also arise in distribution planning and scheduling, and access rights to downloads in case of digital delivery. The software product manager's orchestration focus should be in the following areas: positioning the product into the sales plan, sales plan execution, product specific commitments to customers, handling of customer requirements, deviations from contract terms, deviations from price levels or structures, presales meetings, skills of sales representatives, and alignment of sales metrics with software product management. (ISPMA, 2017)

3.3.4 Service and Support

The service and support organisations are responsible for the services and support offered and provided for the customers as part of the whole product. The service offering typically includes both product-related and consulting services, while support refers to services helping the customers with product-related problems. In addition to support, product-related services include for example education, installation, maintenance, customization, and operations. Software product managers need to have a thorough knowledge of customer problems to identify opportunities, which could be solved with services, and what should be bundled into the product or sold separately. The services may be critical for customer's product adoption and retention specially with products that address complex problems. The software product managers should engage other service professionals to collaborate with the services definition, planning, and delivery. Services can be delivered with internal resources or via partners. (ISPMA, 2017)

Services Planning and Preparation contains the activities related to the development and agreeing of plans concerning product-related service activities within a given timeframe. The plans need to be synchronized with the product strategy and the marketing plan. The preparation typically includes development of the technical basis for the service, estimation of service demand, resource management, skills management, and development of marketing material in collaboration with marketing. (ISPMA, 2016a)

Services Provisioning is the actual execution of the services according to the service plan. Tracking of the service metrics enables taking corrective measures if there are deviations from the service plan. (ISPMA, 2016a)

Technical Support means the activities around fulfilling the maintenance contracts. Support is typically provided in a levelled structure with help desk, technical maintenance, and change team functions. Customers call or communicate the issues related to the software product via digital channels, and the support organisation categorises and manages those in a timely fashion. Additional aspect to technical support which needs to be orchestrated by the software product manager, is the availability of product-related documentation for internal and external purposes. (ISPMA, 2016a)

Marketing Support means activities aiming to assist marketing. Such activities may include for example production and distribution of marketing material, organisation of marketing events, and tracking of marketing activities. (ISPMA, 2016a)

Sales Support means activities aiming to assist sales and sales channels. Such activities may include sales-related customer communication and facilitation, organisation of sales events, providing marketing material on request, and tracking of sales activities. (ISPMA, 2016a)

Sources of conflict in relation to service and support include getting product feedback, sharing the product roadmap, prioritizing features for markets vs. individual customers, training of service personnel, product team support, issue resolution, investments in supportability features, and communication of support issues. The software product manager's orchestration focus considering service and support should be on the following: product-related services and documentation as part of the offering, service execution, skills of service specialists, recurring analysis of incoming service calls, and resource management. (ISPMA, 2017)

3.4 Summary

The ISPMA's SPM Framework maintained by the International Software Product Management Association (ISPMA), structures software product management practices into functional areas, which are further grouped into Core SPM, Participation and Orchestration areas. Core areas include product strategy and product planning as central for software product management. Participation

focuses on strategic management, while the orchestration covers development, marketing, sales and distribution, and service and support activities.

The Core SPM areas of the framework are product strategy and product planning, which define what the software product will be, when and how it will be developed, and how it will be used by the customers. The product manager defines the software product, sets targets, and secures commitment from stakeholders for the evolution of the product. The software product manager is responsible for the decisions taken and takes leadership in managing the different aspects of the product strategy and planning.

Participation includes strategic management, where the product manager participates in the activities, but the responsibility and leadership of the activities belong to the executive management of the company. The executive management sets goals and constraints for the company's product portfolio, and the product manager influences the company's strategy. This area can be considered to represent the interface between software product management and the company's executive management.

Orchestration contains four functional areas from which the software product manager depends on in realizing the product vision and executing the product strategy. Development function implements the software, while marketing identifies and wins the customers. Sales and distribution close the deals, nurture the customer relationship, and generates revenue. Service and support facilitate the product use over its lifetime. The software product manager could be described as a "internal customer" and orchestrates these different company functions of development, marketing, sales and distribution, and service and support, but delegates responsibility and navigates the conflict-filled organisational structure with an influencer leadership mindset.

The implementation guidance for ISPMA's SPM Framework avoids trying to present a one-size-fits all process framework, but instead gives high level classification for product activities and how they should be planned in the software product manager's workplans. For the core SPM activities, section 3.1.1 presented grouping of activities related to product strategy under continuous, periodic, and triggered categories. Similar grouping for product planning was presented in section 3.1.2. For participation and orchestration framework areas such guidance is not presented, but potential conflict and focus areas are listed for each other company function in section 3.3. Therefore, it is relevant to research further how such processes could be implemented and improved in practice. In the next chapter, potential approaches for process improvement in software product management are investigated.

4 PROCESS IMPROVEMENT

What is a process? Davenport (1993) defines that a process is a "specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs: a structure for action" (Davenport, 1993, pp. 5). The process structure puts emphasis on how the work is done, instead of what is being done. The "structure of action" is key of success, since unless participants in the work activities can agree on it, it is very difficult to improve processes. (Davenport, 1993)

Process structure is considered separate from the organisation's hierarchical structure. The organisation structure is typically a view of responsibilities and reporting relationships, while the process structure is a more dynamic view of how the organisation delivers value. Processes have measurable cost, time, quality, and customer satisfaction, while the organisation itself does not. With these measurements it is possible to evaluate the need for improving the processes further. Processes require clearly defined owners that are responsible for the design, execution, performance, and development of the processes, which should always include customer satisfaction as the process output measurement. The ownership of the process rarely matches the existing organisational boundaries of power and authority, and it requires extra consideration when implementing processes. (Davenport, 1993).

The issue of fitting together organisation and its strategy through processes, structure, and technology has been researched extensively for the last 40 years. The increasingly changing economic environment has increased the interest in improving the organizational business processes to enhance performance of the companies. The success in implementing organisational changes depends on the quality of the implementation process. While the number of processes on a company can be large, the usual focus on process improvement is on a small number of key processes on a company level, since changing all identified processes at the same time is a recipe for failure (Trkman, 2010).

According to Pollack and Pollack (2015), Kotter's (1996) eight-stage process of creating a major change is one of the most well-known approaches to organisational transformation, change management leadership, and success in

change management. The eight stages of the Kotter's process are as follows: 1) establishing a sense of urgency, 2) creating the guiding coalition, 3) developing a vision and strategy, 4) communicating the change vision, 5) empowering broad-based change, 6) generating short-term wins, 7) consolidating gains and producing more change, and finally, 8) anchoring new approaches in the culture (Kotter, 1996). Pollack and Pollack (2015) found out that applying the process in practice is significantly more complex than theory.

Software products cut through across the company's value chain, and to align the product with the whole company, the software product manager must coordinate company-internal stakeholders, processes, and activities. The success of the software product depends on the organization's capability to work with a set of dependent competences, practices, and processes related to the software product. (Fricker, 2012)

Software processes as human-centered activities are inclined to unexpected or undesired behaviour and performance. Therefore, software processes need to be continuously evaluated and improved, to ensure fulfilment of the requirements set for the software by customers and internal stakeholders within the software organization. (Unterkalmsteiner et. al, 2012)

4.1 Software process improvement

Software Process Improvement (SPI) has been a popular approach to delivering improvements in software products (Humphrey, 1989), and it became well-known approach not only in the industry, but also by researchers in the 1990s (Hall, Rainer & Baddoo, 2002). Software Process Improvement methodology can be defined as "definitions of sequence of tasks, tools and techniques to be performed to plan and implement improvement activities" (Aysolmaz & Demirörs, 2011). Many companies have executed either formal or informal SPI programs, which generally have delivered significant benefits (Hall et al., 2002).

In a systematic mapping study by Kuhrmann, Diebold and Münch (2016), research articles from the last 25 years concerning software process improvement were categorised. The growing trend of research focus is clear, but software product management as a subject has been on very low focus compared to research focusing on general improvement, or new trends such as agile/lean methodologies.

In 2009, a group of experts in software process improvement started working on defining the core values and principles for process improvement. As a result, they published a SPI manifesto in 2010 containing three core values and 10 principles. The three core values in the manifest are people, business, and change. The "people" value means that the software process improvement effort must involve people actively and affect their daily activities. The second value "business" means that the software process improvement activities must be done to make business successful, instead of compliancy reasons. The final third value "change" means that the software process improvements strive for

change, and not continuing as things are done today. The 10 different principles in the manifesto are organized around the core values: four principles related to people, and three principles related to business and change. The principles should assist software process improvement practitioners in decision making. (Pries-Heje & Johansen, 2010)

The most recent iteration of the original three-word SPI is much more extended version: S(5)P(2)I(3) – which means System, Software, Services, Safety, Security, Process and Product Improvement, Innovation, and Infrastructure. However, the values and principles have not been updated accordingly and updates are expected. (EuroAsia SPI, 2019)

Over the decades, different software process improvement models, standards and recommendations have been developed to help software organisations to manage the software development processes (Bekkers et al., 2012, Khan et al., 2017). Unterkalmsteiner et. al (2012) have classified software process improvement initiatives into three categories: frameworks, practices, and tools.

Frameworks include for example Capability Maturity Model (CMM) and its successor Capability Maturity Model Integration (CMMI), which both include five maturity levels, key process areas, and structured practices for process assessment and improvement (Khan et al., 2018). Level 1 of CMM and CMMI refer to the lowest maturity state of an organization while level 5 is the highest maturity level (Khan et al., 2018). The frameworks are maintained by Software Engineering Institute (SEI) among other minor frameworks (Unterkalmsteiner et. al, 2012).

A joint workgroup between International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) developed the ISO/IEC 15504 standard for process improvement within SPICE program (Khan et al, 2018). The SPICE stands for "Software Process Improvement and Capability Determination", which is a framework consisting of five capability levels, processes, and capabilities (Khan et al., 2018). Large parts of ISO/IEC 15504 have been replaced by ISO/IEC 330XX standard series in 2015 (ISO, 2019). Other ISO standards can also be related to software process improvements – for example, ISO 9000 is generally used in assessing quality of established management systems of any kind (Khan et. al, 2017).

Unterkalmsteiner et. al (2012) have also listed several other frameworks which could be applicable for software process improvement such as Six-SIGMA, PSP, TSP, QIP, BOOTSTRAP, TQM, IDEAL, and PDCA. Many of the most utilized frameworks listed above address parts of software product management related aspects, however research have shown that companies find those models too heavy to use and implement, expensive and time-consuming (Bekkers et al., 2012). For that reason, a model specifically for software product management is presented in section 4.4.

Dekkers and Forselius (2007) have found in their study concerning scope management, that also software process improvements do not always succeed even with the best of intentions. Furthermore, they presented success criteria by Willman (1996) for any organisational change with results when any of the

components are missing. One can find similarities between the Kotter's (1996) eight stages and the success criteria by Willman (1996). The table below (table 7) presents the five success criteria on top and the results on the right if any of the components is missing, indicated by lack of "+" in the table. For example, if the pressure for change is missing, the organisational change effort will fail due to disinterest to the change.

TABLE 7 Model for organisational change (Willman, 1996).

Pressure for change	**Leadership and vision**	**Capable people**	**Actionable first steps**	**Effective rewards**	**Results**
+	+	+	+	+	Successful implementation
	+	+	+	+	Disinterest
+	+	+	+		Evaporation
+	+	+		+	Frustration
+	+		+	+	Disengagement
+		+	+	+	Dissolution

There have been extensive studies on what are considered as factors influencing the success of software process improvement. Success factors and barriers have either positive or negative impact on human related aspects of software process improvement (Khan et al., 2017). In a study by Goldenson and Herbsleb (1995), six success factors and five barriers for software process improvement were found. The results also suggested that "process maturity does indeed pay off in better product quality, ability to meet schedule commitments, and other indicators of organizational performance" (Goldenson & Herbsleb, 1995, pp. 1).

Empirical research by Niazi, Wilson and Zowghi (2006) defined seven factors, which can be considered critical for the success of software process improvement. Galinac (2009) went extensively through five different studies to identify 20 different factors influencing software process improvement success. In one of the more recent studies, Khan, Keung, Hussain, Niazi, and Tamimy (2017) have identified total of ten human related success factors and eight barriers for software process improvement success. Furthermore, they defined that five of the success factors and three of the barriers are critical. The barriers were more extensively mapped in an extensive systematic literature review by Khan, Keung, Niazi, Hussain, and Shameem (2018), where 20 barriers – or challenges as renamed by them – were found, and four of them were defined as significant. The combined list of five critical success factors and seven barriers by Khan et al. (2017) and Khan et al. (2018) are described in the next section.

4.2 Critical success factors and barriers

The first critical success factor is organizational *management commitment* to the software process improvement activities, which plays a significant role in successful implementation. The management commitment needs to be supported by financial backing and show continued involvement to the effort. *Staff involvement* is important during the process definition of software process improvement activities to make them feel part of the improvement program and reduces resistance to the change. Staff involvement can be taken to the extent where the organisation staff own, participate, and achieve the process improvement activities. The effect of having *skilled human resources* as part of the software improvement process has a significant role in the success. The experience in the field of computer science, management, and other related fields for the qualified staff is valuable in the process improvement. It is important to assign specific *roles and responsibilities* to the team members during the software improvement process. The clearly defined roles and responsibilities assist in the success of the process implementation, which may otherwise be confusing. *Leadership of the software process improvement* is vital for the successful execution of the improvement program. The success and acceptance of the software process improvement program mostly depends on the skills of the top management individuals, which should have deep understanding of the software process improvement program, and have the ability to direct and encourage team members in achieving the goals. (Khan et al., 2017)

One of the main barriers for software process improvement is *inexperienced staff*. Teams having skilful and experienced team members succeed better. As an example, lack of knowledge of the process improvement program can cause problems. *Staff turnover* during the software improvement activities is also a critical barrier. If staff members frequently leave the organisation, or are replaced by new staff, if can significantly increase the cost of the process improvement program and cause loss of experienced individuals. *Time pressure* also contributes negatively to the implementation of improvements and is a critical barrier. Too fast decisions made in the software process improvement program to stay on the schedule, may not be in favour of the program goals. In addition, the time pressure is considered as a demotivation factor for the staff in the software process improvement program. (Khan et al., 2017)

Lack of communication is also a significant barrier, since the communication is a vital issue, especially in a global software development environment where the team members are working in a distributed fashion. Lack of face-to-face communication can easily lead to challenges in coordination and control, introduce misunderstandings, and even distrust. *Budget constraints* also act as a barrier regardless of the management commitment, either because for small organisations it is just difficult to budget software process improvement programs, or for those who would need assistance, cannot hire outside consultants. Overall *poor understanding of the software process improvement* concepts either in the man-

agement or staff is also a barrier and is one of the key reasons why software process improvement programs fail. Finally, *lack of implementation standards* is a barrier for successful software process improvement. Lack of tools and standards for process improvements is one of the reasons for failure and may in fact be a superseding barrier even if other barriers are absent. (Khan et al., 2018)

The other success factors identified by Khan et al. (2017) which are not considered as critical, are strong relationship, information sharing, SPI expertise, SPI awareness, and 3Cs (communication, coordination, and control). Both Khan et al. (2017) and Khan et al. (2018) have listed cultural differences, lack of training, and lack of trust as non-critical barriers. Non-critical barriers according to Khan et al. (2018) also includes lack of conflict management, organisational politics, poor technological infrastructure, lack of staff involvement, work pressure, poor organisational commitment, time zone differences, lack of resources, and lack of knowledge sharing.

The success factors and barriers vary somewhat between different studies. Niazi, Babar and Verner (2010) found significant differences between challenges of companies depending on country they are operating in. The critical success factors and barriers by Khan et al. (2017) and Khan et al. (2018) are utilized in this study as the reference framework. The list and classification of the critical success factors and barriers are shown below in a table (table 8).

TABLE 8 Success factors and barriers for software process improvement (adapted from Khan et al., 2017 and Khan et al., 2018).

Type of factor	**Factor**	**Reference**
Success factor	Management commitment	(Khan et al., 2017)
	Staff involvement	(Khan et al., 2017)
	Skilled human resources	(Khan et al., 2017)
	Roles and responsibilites	(Khan et al., 2017)
	SPI leadership	(Khan et al., 2017)
Barrier	Inexperienced staff	(Khan et al., 2017)
	Staff turnover	(Khan et al., 2017)
	Time pressure	(Khan et al., 2017)
	Lack of communication	(Khan et al., 2018)
	Budget constraints	(Khan et al., 2018)
	Poor SPI understanding	(Khan et al., 2018)
	Lack of implementation standards	(Khan et al., 2018)

4.3 Best practices

A best practice can be defined as that technique, method, process, or activity that is more effective at delivering a particular outcome, than any other technique, method, process, or activity within that domain (Camp, 1989). This is in line with Frederick Taylor's (1919, pp. 25) early formative work, which claims that "among the various methods and implements used in each element of each trade, there is always one method and one implement which is quicker and better than any of the rest".

Like the previous sections showed, there has been lot of research done on what are critical success factors for software process improvement. However, as pointed out by Galinac (2009), "even if an organisation is aware of the factors and issues required for success, one of the important prerequisites is to have guidelines on how to achieve those factors". Software process improvement is more successful in teams which follow an implementation strategy including best practices, than in those which do not (Galinac, 2009).

The study by Galinac (2009) also identified 14 best practices related to software process improvement, which all relate to certain success factors. Those best practices are illustrated below (table 9).

TABLE 9 Improvement practices of SPI implementation strategy (Adapted from Galinac, 2009, pp. 1354).

Phase	**Description**	**Best practice**
Opportunity	Opportunity for improvement is determined	Improvement backlog
Content	Goal and scope of improvement are defined	Project specification
Approach	The way of constructing the improvement solutions is defined	Cross organisational team
		Cross process team
		Expert participation
		Core team
		Collaboration tools
		Small deliverables
		Increments
Execution	The improvement solutions are developed	Monitoring
		Iterations and frequent reviews
		Uniform processes
Validation	The solutions are deployed and validated	Deployment plan
		Measurement
Integration	The solutions are integrated with the new revision of processes	

The background for the listing by Galinac (2009) originates from CMMI, where "Organisational Process Focus" is one of the processes in the Maturity Level 3. It has been adapted to better suit the global software development context, and the practices are grouped into a six-phase cycle of software process improvement implementation strategy.

4.4 Process improvement in software product management

For globally operating software product vendors, there are several important aspects to be taken to account with process improvement work. Firstly, shared infrastructure for tools related to the process is critical for process rollout. To facilitate gradual process improvement, it should be done on small increments, as most organisations are not able to adopt large changes in one step. Global involvement is critical, meaning that the involvement and team members for the process rollout must be local, but in a distributed team. And finally, if companies have been acquired, they must be transformed last – it is important to support existing cultures and methods for a while, before moving to evolutionary unification of work methods. (Weerd, Brinkkemper & Versendaal, 2010b)

According to Maglyas et al. (2017), companies adopt software product management processes differently, with one of the main distinguishing factors being the size of the company. This is supported by a theory by Khan et al. (2017), which suggests that success factors and barriers for software process improvement influence larger organizations more, than small or medium-sized organisations.

In a survey done by Product Focus (2019), 42% of organisations state they have processes, which are not really defined or widely used. Alarmingly, almost 25% of respondents are working without any defined process, and only 50% have some process defined, which is generally used.

The adoption of software product management activities should start from choosing the most important activities first. Core SPM activities as are good candidates to start with, as the adoption of all activities instantly may result in a waste of resources without significant improvements. (Maglyas et al. 2017)

Bekkers, van de Weerd, Spruit and Brinkkemper (2010) have developed relevant artefacts related to process improvement specifically in software product management. The artefacts are the Software Product Management Competence Model and Software Product Management Maturity Matrix. van de Weerd et al. (2010a) explain in their paper, that the artefacts were chosen to be built as a focus area-oriented maturity model, because other model types such as CMM, CMMI, and SPICE were experienced as too heavy and complex to implement or even understand.

First of the artefacts, the Software Product Management Competence Model, which is illustrated below (figure 7), gives an overview of all areas which are important in the field of software product management.

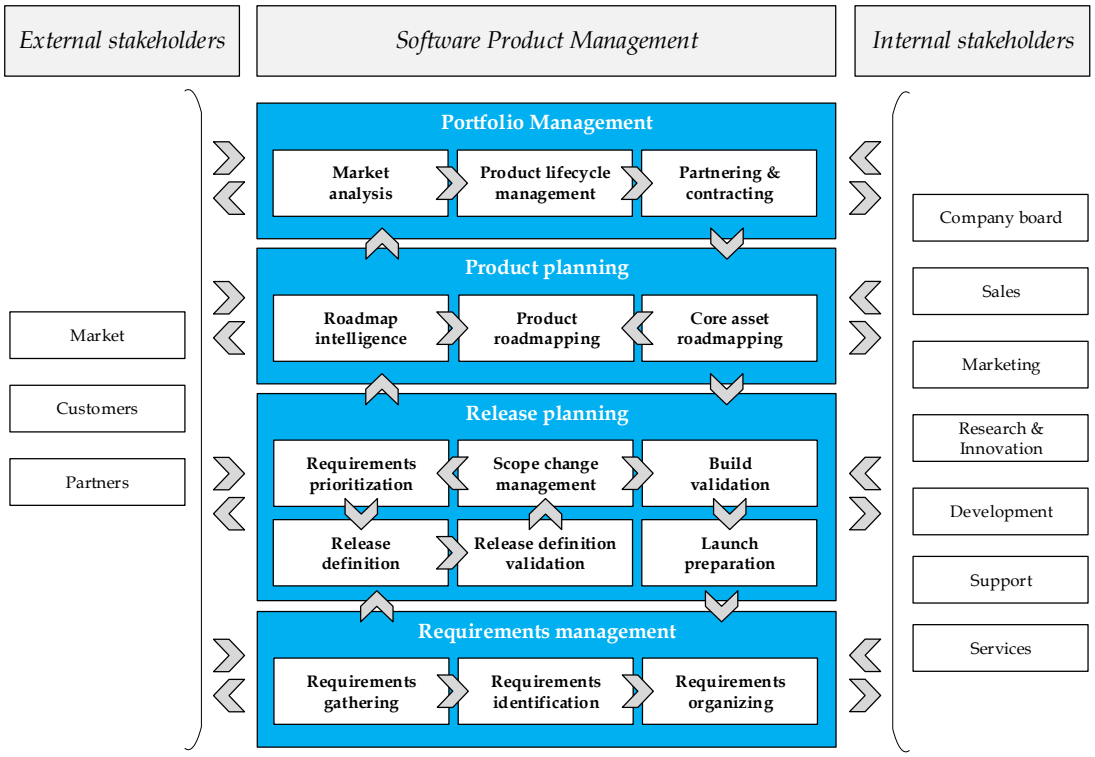

FIGURE 7 Software Product Management competence model (Adapted from Bekkers et al., 2010, pp. 4)

The Software Product Management competence model consists of four business functions in the centre: portfolio management, product planning, release planning, and requirements management. Each of these business functions contain number of focus areas, which are of importance concerning software product management. In addition, the relevant external stakeholders are shown on the left, and the internal stakeholders are shown on the right side. (Bekkers et al., 2010)

The second artefact is the Software Product Management Maturity Matrix, which is structured based on the competence model shown above. Organisations can assess their maturity level based on the matrix, and plan the improvement of their processes, since the model also contains a best practice order of implementation. (Bekkers et al., 2010)

There need for assessments and improvements seems to be real, as current state is of maturity is not great: according to survey by Product Management Festival (2019), only 20% of organisations think they have a mature Product Management organisation. 55% of people working with product management create the processes themselves, 13% do not know there the processes came from, and 12% do not have any processes in place (Product Management Festival, 2019). The Software Product Management Maturity Matrix by Bekkers et al. (2010), is shown below (figure 8).

	Maturity level										
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Requirements management											
Requirements gathering		A		B	C		D	E	F		
Requirements identification			A			B		C			D
Requirements organizing				A		B		C			
Release planning											
Requirements prioritization			A		B	C	D			E	
Release definition			A	B	C				D		E
Release definition validation					A			B		C	
Scope change management				A		B		C		D	
Build validation					A			B		C	
Launch preparation		A		B		C	D		E		F
Product planning											
Roadmap intelligence				A		B	C		D	E	
Core asset roadmapping					A		B		C		D
Product roadmapping			A	B			C	D		E	
Portfolio management											
Market analysis					A		B	C	D		E
Partnering & contracting						A	B		C	D	E
Product lifecycle management					A	B			C	D	E

FIGURE 8 Software Product Management Maturity Matrix (Adapted from Bekkers et al., 2010, pp. 4)

In the matrix, the rows represent the focus areas from the competence model, while the columns represent the maturity level, where 0 is the lowest and 10 is the highest. Each focus area has specific maturity levels, which are indicated by letters A-F. When a focus area is executed at a certain maturity level, it is a called a capability. The distribution of A to F indicates the best practice order of improving each area. An organisation can look at the maturity of both individual focus areas, but also organisation wide by determining the overall maturity based on the lowest achieved maturity level. (Bekkers et al., 2010)

The maturity matrix also contains both built-in intra-process and inter-process dependencies. Intra-process dependencies mean, that one capability within a certain row is dependent of another capability on the same row, for example all B capabilities depend on A capabilities on the same row. Inter-process dependencies refer to a dependency between a capability in a certain row to a capability on a different row in the same business function. (van de Weerd et al., 2010a)

The Software Product Management Maturity Matrix has been evaluated in 62 case studies during a three-year period. Most of the assumptions in the matrix were confirmed, and the matrix has been further adjusted and proven to be a good instrument in assessing company's maturity in the field of software product management. (Bekkers et al., 2012)

The maturity of the product management processes is an important factor for the job satisfaction of the employees. According to Product Management Festival (2019), "product management maturity" was ranked as the most desired characteristic (35,5% of respondents) for a Product Management role in 2019.

Ebert and Brinkkemper (2014) have found in their research, that the change process for software product management occurs in three main phases: foundations, pilot, and roll-out. The change process is illustrated below (figure 9) with keywords related to one or more phases.

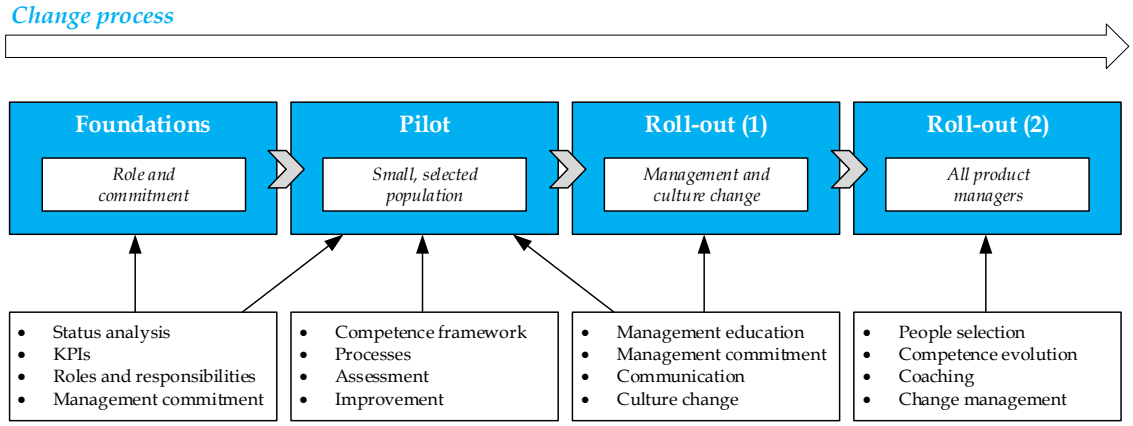

FIGURE 9 Change process for Software Product Management (Adapted from Ebert & Brinkkemper, 2014, pp. 14)

During foundations phase, roles and commitment are established. The pilot is done for a selected population, and finally the roll-out is done to all product managers supported by management and culture changes. This kind of process improvement calls for a professional change management, but it is often underestimated by companies. Many companies still have their primary focus just on training programs, but do not even have the basics in place, such as clear responsibility mapping. (Ebert & Brinkkemper, 2014)

In the same research, Ebert and Brinkkemper (2014) also found success factors and challenges (barriers) specifically for software product management, related to introduction of product management into organisations. Top four success factors are cross-organisational core team, standardized product life cycle, requirements focused on customer value, and implementation of portfolio management and roadmapping. Barriers for success are dysfunctional organization, lack of standardized processes, insufficient requirements, and lack of strategy. (Ebert & Brinkkemper, 2014)

When utilizing the competence model or the matrix for process improvements, situational factors should be considered when assessing the current state of the software product management in an organisation (Bekkers et al., 2012). The next section discusses such situational factors.

4.5 Situational factors

In software product management, there are many additional challenges caused by internal and external factors, which need to be considered when implementing software product management practices.

Situational factors describe the environment, industry, and other key factors related to the software organization's context and what it deals with. The situational factors affect both to the selection of which software product management methods should be applied, and how the processes should be maintained. (Bekkers, van de Weerd, Brinkkemper & Mahieu, 2008)

As discussed in chapter 2, according to Tyrväinen et al., (2008) the software market phases have impact to the software business and software products. Maglyas et al. (2017) and Khan et al. (2017) have further stated that the size of the organisation is one of the factors influencing how companies adopt the practices, and according to Giardino et al. (2016), in software start-ups, the traditional software process activities can be even discarded in favour of growth. Bekkers et al. (2008) have found as many as 27 different situational factors, which are relevant to software product management in five categories: Business unit characteristics, Customer characteristics, Market characteristics, Product characteristics, and Stakeholder involvement.

Each situational factor influences different aspects of the software product management with a different weight: some factors matter more than others in a specific software product management process area, or in several areas. According to Bekkers et al. (2008), situational factors within the Customer and Product characteristics categories have the most significant impact in all software product management areas. Individual situational factors such as customer loyalty, customer satisfaction, product age and product lifetime ranked as the most impactful success factors. (Bekkers et al., 2008)

In a later research by Bekkers (2012), the situational factors were further refined into a Situational Factor Effects Catalog, containing 25 situational factors, and the kind of effects they have for daily practice, capability implementation, or method selection. In contrast to the earlier research, the creation of the catalogue showed that the most impactful situational factors would be in fact found in the Stakeholder involvement category. The most influencing individual situational factor was customer involvement, which had a lot of different influences to daily practices, capability implementation and method selection. In any case, each situation factor was found to have at least one practical effect, and half of the factors had multiple types of effects. The summary of the situational factors is shown in the table (table 10) below.

TABLE 10 Situational Factor Effects Catalog (Adapted from Bekkers, 2012, pp. 174).

Situational Factor	Category	Daily Practice	Capability Constraints	Method Constraints
Development philosophy	Organisation	X		X
Size of organization	Organisation	X		X
Customer loyalty	Customer			X
Customer variability	Customer	X		X
Number of customers	Customer			X
Type of customers	Customer	X		X
Market growth	Market			X
Release frequency	Market			X
Sector	Market	X		X
Market size	Market			X
Standard dominance	Market	X		
Variability of feature requests	Market			X
Number of localizations	Product	X		X
Defects per year	Product	X		X
New requirements rate	Product			X
Number of products	Product		X	X
Product age	Product			X
Product lifetime	Product	X	X	
Product size	Product	X		X
Product tolerance	Product			X
Company policy	Stakeholder			X
Customer involvement	Stakeholder	X	X	X
Legislation	Stakeholder	X		
Partner involvement	Stakeholder	X	X	X

4.6 Summary

Processes are specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs. Dynamic process structures emphasize on how the work is done, instead of what is done. Processes are separate from the organisation's hierarchical structure, are led by defined process owners and focus on how the organisation delivers value in a measurable way.

Improving and implementing cross-functional business processes have been researched extensively in search for enhancements in companies' economic performance. Similarly, software products also cut through companies' value chain, and the success of the software product management depends on the human-centered processes, which are inclined to have unstable behaviour.

Software Process Improvement (SPI) is a popular approach to deliver improvements in software products which contains definitions of sequence of tasks, tools, and techniques to be performed to plan and implement improvement activities. Much research has been done in software process improvement both generally, and on different aspects such as process improvement frameworks CMMI and ISO/IEC330XX.

The software process improvement occurs in several phases, and there are phase-specific best practices, success factors and barriers to consider. Successfully implementing process improvements require management commitment, staff involvement, skilled human resources, clear roles and responsibilities, and professional change leadership. Inexperience of staff, staff turnover, time pressure, lack of communication, budget constraints, poor process improvement understanding, and lack of implementation standards act as barriers for the process improvement.

Process improvements in software product management, however, have been a relatively low focus area until recently. Some of the approaches in SPI have been found to be too complex for software product management, and more fitting models have been created. The Software Product Management Competence Model and Software Product Management Maturity Matrix assist in evaluating the current state of a software organisation's product management maturity level and planning on to which process areas the improvements should focus on, and in what order.

Change process for software product management occurs in several phases – foundations, pilot, and rollout – and each phase has its own focus areas. The success factors and barriers for software product management process improvement have similarities to those generally found for SPI, but some of them focus more on the resulting processes, than the process improvement itself.

The context and setting of the software organisation have an impact to which processes should be improved, how they should be improved, and which methods should be selected. Such internal and external situational factors must be included in the status assessment when undertaking software product management process improvement work, as there are no two companies which are exactly alike.

5 RESEARCH METHOD

The aim of the study was to understand what the best practices have been in improving software product management processes by ISPMA's SPM Framework practitioners, and what kind of benefits have been achieved by applying the framework. The research approach for this study was mixed methods approach, which combined a quantitative survey and qualitative semi-structured interviews.

The first section introduces the used research methodology, factors impacting to the selection of the chosen approach, and background for the chosen research methods. The second section describes the data collection process, and the third section describes the analysis of the collected data. Finally, the fourth section discusses the reliability and validity of the research.

5.1 Choosing the methods

According to Hirsjärvi, Hurme and Sajavaara (2009), it is difficult in practice to define a clear distinction between quantitative and qualitative research, and they are often seen as complementary methods and approaches. Venkatesh, Brown and Bala (2013) have stated that mixed method research has been described as the third methodological movement after the quantitative and qualitative methods.

Research strategy in mixed methods utilizes multiple methods, which can be either more than one research method, or more than one worldview (i.e. quantitative or qualitative). The methods are used either concurrently (i.e., independently) or sequentially (e.g., later methods build on the results of the previous methods) to understand the target of the study. (Venkatesh et al., 2013)

According to Venkatesh et al. (2013), there are three major strengths of mixed methods research. Firstly, mixed methods research can address both confirmatory and exploratory research questions simultaneously. Secondly, two data collection approaches can help information systems researchers make bet-

ter and more accurate inferences on the studied phenomenon. For example, qualitative interviews can provide depth in the research by gaining deep insights, while quantitative surveys can bring breadth to the study by gathering data from many participants. Thirdly, mixed methods research opens the possibility of gathering a greater variety of contradictory and/or complementary views on the same subject. This can be valuable since it may lead into re-examination of the conceptual framework and assumptions of the study subject.

Before starting a mixed methods research, the appropriateness of utilizing such approach must be carefully considered (Venkatesh et al., 2013). Venkatesh et al. (2013) describe a comprehensive list of purposes for using the mixed-method approach instead of other approaches. The seven purposes are complementarity, completeness, developmental, expansion, corroboration (or confirmation), compensation, and diversity.

In this study, mixed methods were primarily used for complementarity and completeness purposes, so that a complete picture of the phenomena could be created including complementary views of varying angles and depths. The methods were used in sequential fashion by conducting the survey first and interviews later, which also enabled taking advantage of the developmental purpose of the mixed methods. Finally, mixed methods were also used for compensation purposes to improve the research validity and reliability, since it was suspected that the sample in the quantitative survey might not be statistically representative, or there might be challenges of reaching suitable interviewees.

Mixed methods research occurring in multiple phases can be difficult to understand by textual explanation only (Ivankova, Creswell & Stick, 2006). Drawing a graphical representation helps both the researcher and the readers to comprehend the study and the procedures used within (Ivankova et al., 2006). Such visual model about the phasing of this research is illustrated below (figure 10).

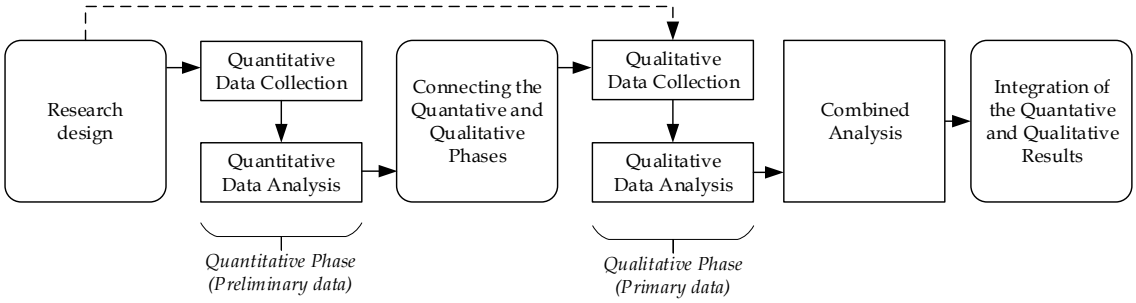

FIGURE 10 Phasing of the research

The first phase was research design, which included both quantitative and qualitative research design. It was followed by a quantitative phase, where preliminary data was collected via survey method and analysed. After that, the qualitative phase of the research was conducted with additional input from the quantitative phase. Qualitative phase consisted of qualitative data collection via semi-

structured interview method and data analysis, producing primary data for the research. Preliminary data and primary data were analysed together in the combined analysis phase before the final phase, integration of the quantitative and qualitative results.

The proceedings of the quantitative and qualitative data collection and analysis are described separately for clarity in the following sections. Before delving deeper, the selected research methods for those phases, the survey, and the interview, are shortly explained as research methods in the following subsections.

5.1.1 Quantitative survey

In this thesis, survey research was chosen as the quantitative research method and was planned to be used primarily for descriptive purposes, and as preliminary data for the qualitative phase of the research. It was intended to provide a backdrop or overview of the ISPMA's SPM Framework practitioners, and the status of software product management process implementation within different enterprises.

Survey research is one of the most central methods used in quantitative research (Hirsjärvi et al., 2009). The method has a long historical tradition (Hirsjärvi et al., 2009), and according to Venkatesh et. al (2013), surveys are also the most used method for the quantitative strand of mixed methods research in the field of information systems science.

Survey research means such forms of data collection, where data is usually collected in a standardized way, and the target persons form a sample of the total population. Questions in the survey are asked from all respondents exactly in the same way, and the collected data is usually analysed quantitatively. (Hirsjärvi et al., 2009)

Surveys can be used to collect information about facts, behaviour, information, values, attitudes, beliefs, and opinions. Typically, also demographics about the respondent is collected, such as gender, age, and education. The questions in the survey forms are usually one of three types: open questions, multiple choice questions, or scale questions. With open questions, the respondent can provide a freely formatted response, while with multiple choice questions one must choose from a predefined list of options. Variations can also occur, such as multiple-answer variants, or combining multiple choice selections with an open answer option. Scale questions often utilize questions as statements, for which the respondent must choose a suitable degree of opinion from a multi-step scale, such as the Likert scale. (Hirsjärvi et al., 2009)

Survey research can be used for exploration, description, or explanation purposes. Exploratory surveys can be useful as independent researches, or more often, as preliminary phases for other studies. Descriptive surveys can be used to create a picture about current state and point out areas where further study may be useful, while explanatory surveys focus on testing theories and directional relationships between variables. (Pinsonneault & Kraemer, 1993)

One of the greatest benefits of utilizing surveys is, that it can be used to gather large amount of empirical data: many persons can be included in the research and many questions can be asked with a relatively low effort from the researcher. The collected data can be quickly converted into a form which can be analysed with computer-assisted methods, typically supported by ready-made statistical analysis and reporting formats. (Hirsjärvi et al., 2009)

There are also downsides to survey research. Data collected in a survey research can be deemed as superficial and creating good survey forms with properly formatted questions requires time and skill. There can be also several unknowns about the respondents. For example, inability to evaluate how seriously the respondents have taken the survey, how understandable the survey question has been, and how well-versed the respondents have been about the subject of the survey. Survey researches can also have a low response rate compared to the questionnaires sent. (Hirsjärvi et al., 2009)

5.1.2 Qualitative interviews

In this thesis, semi-structured interview was chosen as the qualitative research method to produce primary research data. It was chosen due to the openness and improvisation possibilities, which allow delving deeper into interesting lines of research based on unexpected responses. It was expected that the research would uncover such best practices and benefits, which would be hard or impossible to find with too superficial methods.

Interviews are the main method in qualitative research in general (Hirsjärvi et al., 2009), and it is also the case in mixed method studies for the qualitative research in information systems science (Venkatesh et al., 2013). According to Hirsjärvi et al. (2009), interview methods can be separated into three main types: structured interviews, theme interviews, and open interviews. Structured interviews have a complete script of questions, which is prepared beforehand and there is no room for improvisation, either in the formatting of the questions, or in which order they are presented (Hirsjärvi et al. 2009). According to Myers and Newman (2007), structured interviews are often used in situations when the interviews are not necessarily conducted by the researcher. With an open interview method, the interviewer finds out the thoughts, feelings, opinions, and beliefs as they come during the natural flow of conversation, without a predefined interview structure at all (Hirsjärvi et al. 2009). These interviews take a lot of time, require good interviewing skills, and require multiple sessions of interviews with the same research subject (Hirsjärvi et al. 2009).

Themed interviews (Hirsjärvi et al., 2009), which are a form of semi-structured interviews, fall between the structured and open interviews so that they have an incomplete script, which leaves room for improvisation during the interview (Myers & Newman, 2007). The researcher may have prepared some questions beforehand (Myers & Newman, 2007), or only the interviews themes are known (Hirsjärvi et al., 2009). The interviewer usually asks a question, and then more improvised questions are followed based on the reply of the inter-

viewee (Adams, Khan & Raeside, 2014). The interviewer is the researcher themselves, or other integral member of the research team (Myers & Newman, 2007). In any case, the interviewer must ensure that all questions and interview themes are covered (Myers & Newman, 2007), regardless of the exact question format or order of the semi-structured interview (Hirsjärvi et al., 2009).

The main benefit of interview as a data collection method is flexibility, as the researcher can adjust the data collection according the interviewing situation and the interviewee, especially with the themed and open interview methods. Another benefit is that usually the interviewees can be easily involved into the research, and they can be later reached if there is a need to complement the data. (Hirsjärvi et al., 2009)

The interview methods are not without disadvantages. The interviews take time, both in planning and conducting them. Interviews can also be a source for error in the data caused by the interviewer, interviewee, or the interview situation itself. For example, the interviewees may give socially desirable answers, which makes it crucial for the interviewer to interpret the answers correctly. The collected data is also context and situation dependent, which must be considered in the analysis, so that the generalization of results is not exaggerated. (Hirsjärvi et al., 2009)

5.2 Data collection

As described previously, data collection methods chosen for this study were quantitative survey and qualitative semi-structured interviews. This section describes in detail how the data collection was planned and executed.

Part of research design is to define the population and representative sample, which can be used as basis for generalizations and transferability (Hirsjärvi et al., 2009). Sampling can be defined as the process or technique for selecting a suitable sample, which can represent the characteristics of the whole population (Adams et al., 2014). The selection process must consider the size of the sample, its statistical justification, methods of sampling and finally, time and cost (Adams et al., 2014).

Sampling methods are often categorized by probability: in probability sampling methods the likelihood of being selected to the sample is equal, while in non-probability sampling methods the selections are based on personal judgement (Patton, 1990, Adams et al., 2014). While different sampling methods have a lot of differences, they are all designed to provide a sample that will answer the research questions, and they are all concerned about generalizability or transferability (Teddlie & Yu, 2007). In mixed methods research, the sampling strategies may utilize all different types of methods – in fact, the researcher's ability to creatively combine different sampling methods is one of the defining characteristics of mixed methods research (Teddlie & Yu, 2007).

The population for the research was software product management professionals globally, who are utilizing the ISPMA's SPM Framework. To be able

to put the research into this context, it was seen important to learn more about the practitioners and the organisations via demographics. The best source for the information was the ISPMA organisation who maintain the framework, the related certification programs, and a membership database about the world-wide certificate holders. Therefore, it seemed natural to approach the organisation itself to gain access to the practitioners.

Since the whole population of the study could still not be exactly determined and most likely reached, a non-probability sampling approach was chosen. The main sampling technique was voluntary response sampling – a form of convenience sampling – which was combined with criterion sampling and snowball sampling to reach as representative sample as possible.

In convenience sampling, the sample is selected via practical criteria such as easy accessibility, availability at given time or willingness to participate to the study (Etikan, Musa & Alkassim, 2016). In criterion sampling, all cases which meet some predetermined criteria are selected to the sample (Patton, 1990). One form of criterion sampling is also identifying cases from quantitative questionnaires for in-depth follow-up (Patton, 1990). In snowball sampling, information-rich key informants are found via recommendations from well-situated people (Patton, 1990).

Mixed methods were employed sequentially in this research, and in such cases the methodology and results from the first strand of study usually inform the methodology utilized in the second strand. This was utilized also in sampling, where the sampling frame in the qualitative interviews was a subsample of the quantitative survey's final sample. This is aligned with the typical design in sequential mixed methods sampling as defined by Teddlie and Yu (2007).

Contacting the ISPMA organisation was found to be easy due to the researcher's existing contact network. First contact was established by email in November 2019, and the research plan was discussed over Skype in December 2019 with the representative of the ISPMA. ISPMA agreed to support in the research by providing access to the email list for contacting potential respondents and giving away a discount ticket to an upcoming SPM Summit conference to one randomly selected and willing research participant. The original research plan created in 2019 included also contacting survey respondents and potential interviewee candidates via personal contacting at the SPM Summit 2020 (planned for 24. – 25.3.2020 in Frankfurt) to increase the sample sizes. However, the international outbreak of COVID-19 coronavirus made this plan impossible since the event was postponed. For the sake of clarity, it should be stated that this research was not done for ISPMA, has not been guided by ISPMA and there was no compensation whatsoever for the researcher. The following subsections describe the data collection for the quantitative survey and qualitative interviews further.

5.2.1 Quantitative survey

As software product managers are busy people with their many responsibilities, and they are located worldwide, using a standardized survey was the most practical approach for doing the quantitative research. The survey form was planned to be a web form, and the link to it with invitational would be sent via the ISPMA members email distribution list. Therefore, the sampling technique used in the survey research was voluntary response sampling, based on ease of access.

Due to the international nature of the research and subject, the question form and the questions were written in English. Google Forms was chosen as the technical platform for executing the web survey, the main reasons being ease of use, flexibility, and cost. Themes and subjects for questions for the survey were developed using the literature review and theories as background. The questions were internally mapped against one or more of the following contexts: Software Product Management, Situational Factors for Software Process Management, ISPMA / SPM Framework, Software Process Improvement, and Success Factors in Software Process Improvement. In addition, a few validity control questions were identified and added.

After the questions had been planned in content, they were organized into eight sections visible for the respondent: Invitation, Research notification, Personal background, SPMBoK / ISPMA Framework, Success Factors, Process Maturity, Company and product background, and Summary. Each of the sections contained short background description about the section's topic and up to 11 questions per page. Majority of the questions were multiple choice questions, accompanied with some open answers and scale questions. The last section of the survey also contained the possibility to participate in a draw for a discount ticket to an ISPMA event, and possibility to express interest in attending an interview. The choices were not related, but both required the respondent to enter their otherwise optional contact details (email address).

The survey was then tested to ensure the clarity and flow of the questions. This pre-testing was first done with two of the researcher's colleagues, who had software product management experience. Finally, a senior representative of ISPMA also took the survey for testing purposes. The test subjects answered the survey as the actual respondents would and were also given a chance to provide feedback after completing the survey. After small adjustments based on the pre-testing and feedback, the survey form was completed to its final form.

After all preparations were done, an invitation email was formatted and sent to ISPMA, who forwarded it to their email list consisting of 492 certified ISPMA members. The number of all certified members is roughly twice the number, but they were unfortunately not reachable by email distribution. The invitation letter and the first page of the survey can be found in the second appendix (appendix 2). The third appendix (appendix 3) shows the final structure of the survey with the sections, questions, and type of questions. Response op-

tions for multiple choice questions are not detailed in the appendix, but they can be indirectly seen as part of the survey results in chapter 6.

A total of 25 responses were received between 5.3.2020 and 20.3.2020 after two reminders sent out to the mailing list. An additional response was received on 22.4.2020 after one more person was interested to participate due to a snowball effect (reference from another respondent), increasing the total to 26 responses.

5.2.2 Qualitative interviews

The target of the survey was to produce representative demographics of the worldwide ISPMA's SPM Framework practitioners, and views on what kind of practices are or could be used when implementing or improving processed based on the framework. To find our more details on the subject, all survey respondents were also asked whether they would be available for follow-up interviews. The sampling frame for the interviews was therefore a subset of the quantitative survey's final sample. The sampling technique for the qualitative interviews was voluntary response sampling combined with criterion sampling and snowball sampling.

The purpose of the interviews was to get more detailed information about the best practices partially covered in the survey, and to increase the breadth of the research by asking questions outside of the survey scope. In preparation to the interviews, a structure for the interview sessions was developed. It was developed so that the questions were either deeper questions related to survey themes, or questions related to new themes not asked in the survey. For example, the success factors for process improvement were detailed based on information gained in the survey, but benefits for process improvements was a completely new theme for the interviews. All themes were backed up by references to supporting theories, and the questions were also classified preliminary whether the answers would provide background, opinion, or knowledge type of data.

The questions were then grouped into themes and ordered within the themes for discussion structure. Four larger themes emerged: "State of the ISPMA's SPM Framework implementation", "Status assessment, guidance and tools for process improvement", "Ways of ensuring success in the process improvements", and "Realized and expected benefits". In addition, an additional theme was an introductory part, where clarification questions were asked about the interviewee and the company they are employed at. As all the interviews were conducted in the English language, the questions were also written in English.

Microsoft Teams was chosen as the technical platform for conducting the interviews, mainly for ease of access, recording capability, and cost reasons. To assist with the flow of the interview, supporting presentation material was created with Microsoft PowerPoint for the interview session. It consisted of short

introductory slides and supporting slides about the some of the predefined questions in the interview.

The interview structure and suitability of the chosen technical solution was then tested to ensure the clarity and order of questions, timing, and interview recording. Testing was done with one of the researcher's colleague who had earlier participated to the survey testing. The interview was conducted in the same way the actual interview would go, with possibility to give feedback about the interview. After small adjustments based on the testing, the interview phase was ready to begin with actual subjects.

An invitation email was sent to the survey respondents, with link to booking a one-hour interview slot suiting to their calendars. A calendar booking service called Doodle was used to provide the timeslots and avoid potential time conflicts. First one-to-one interview was conducted over Microsoft Teams on 7.4.2020, and the last one was done on 24.4.2020.

The interview sessions at scheduled times were opened with a short introduction from both the interviewer and interviewee. Then a permission for recording was asked, and the actual interview started. First, two introductory presentation slides were shown to highlight about the confidentiality of the interview and remind about the content of the ISPMA's SPM Framework. After that, the interview went on along with the predefined structure so that each theme was covered. The presentation, which included some of the interview questions under each theme, was used throughout the interview to support both the interviewer and interviewee to focus on the topic at hand.

For the most interviews, the order of the themes and questions within the themes was the same, but there was some variation. In many cases some of the predefined questions were handled in the interview earlier than in the structure and were not necessary to repeat. Regardless, all planned themes and predefined questions handled in all the interviews in one form or another. In most of the interviews there was at least a few instances when the discussions brought up interesting new topics, and the questioning went deeper into the topic outside the predefined questions, before returning to the planned track. Fourth appendix (appendix 4) shows the planned structure of the interviews with themes and pre-defined questions. The interviews ended with an open question whether anything else came up regarding the topic. After the recording was shut down, for the purposes of snowball sampling, the interviewees were also asked if they could refer to anyone who might be interested to participate as well. Finally, there was short concluding discussion and closure of the meeting.

A total nine (9) of the survey respondents initially volunteered as interview subjects. Two of the volunteers were never reached, but seven of the initial volunteers were successfully interviewed. One additional person was interviewed after reference from another interviewee, bringing the total persons interviewed to eight (8) persons. It is worth to note that all the respondents participated on their personal interest about the topic as there was no compensation promise for participation. The duration of most interviews was between 45 minutes and one hour, according to the recordings done with Microsoft Teams.

5.3 Data analysis

In an empirical study, conclusions about data can often be done only after preliminary analysis, which can be separated to three steps: checking, complementing, and organising of data (Hirsjärvi et al. 2009). In this research, the data was initially collected sequentially to two separate data sets: the survey results (preliminary data) and the interviews (primary data). This section describes how the research data was analysed, first per data set and then in combination.

5.3.1 Quantitative survey

The quantitative survey data was originally collected with Google Forms survey, which stored the responses directly to a linked Google Sheets spreadsheet. The data was exported to Microsoft Excel format, and details of the individuals were removed from the dataset.

Before starting detailed analysis of the collected survey data, responses must be checked for consistency and completeness. In some cases, it is also necessary to convert ordinal and nominal data from textual data to numerical scores, or code response categories for open questions prior to analysis, so that quantitative data can be analysed. (Kitchenham & Pfleeger, 2003)

Missing data and non-responses were excluded completely from the results, since partially completed surveys did not show up in the data at all. To some questions, it was possible to give open-ended answers or fill in "other" answer option for multiple-choice questions. Open-ended answers were collected into a separate document, and open answers to multiple-choice questions were included in the answer records as new alternatives. In some cases, open answers for multiple choice questions were combined and adjusted for variants, for example in some cases where there were typing errors, but the answers by different respondents meant the same.

Descriptive statistics help in understanding the data and summarize the key issues, and such summary statistics can be visualized either graphically or in tabular form (Adams et al, 2014). The following methods of statistical analysis were used for the data collected in the survey: frequencies, percentages, weighted arithmetical mean, and diagrams. The data analysis was done utilizing Microsoft Excel. In the analysis, descriptive statistics were created to multiple spreadsheets representing each variable in the questionnaire based on all respondent's data. In addition, a few answers to open-ended questions which were not additional options for multiple choice questions, were thematically grouped, and left for qualitative analysis.

5.3.2 Qualitative interviews

The challenge of the qualitative analysis is to make sense of huge amounts of collected data, reduce it to understandable size, identify significant patterns and

finally formulating a communicable framework with the core of the findings (Patton, 1990). For interviews, there is no substitute for actual quotations spoken by interviewees, and recording the interviews is essential (Patton, 1990). The interviews were recorded with Microsoft Teams to ensure that the discussions could be analysed afterwards. Separate interview notes were also taken to a text file, to remind about key points of interest for further analysis. After each interview, the recording was immediately technically checked that it was successful and stored securely for transcribing.

The most desirable data to obtain from interviews is full transcription of the interviews (Patton, 1990). As the period after the interview is "critical to the rigor and validity of qualitative inquiry" (Patton, 1990, pp. 352), the transcription of the interviews was done on the same day, or latest one day later, to ensure that the material was fresh in mind, to cross-check with the notes, and to enable start of the analysis as soon as possible. The transcription was done using a Sonix transcription service, which pre-processed the recorded audio files, which were then verified and corrected by the interviewer against the same audio files. During the verification, all sensitive details such as names of persons or companies, were redacted from the transcriptions. The output of the transcriptions was otherwise in verbatim Microsoft Word text file per interviewee.

As each transcription completed, content analysis was started. During content analysis, the data is coded and categorized so that primary patterns in the data can be identified (Patton, 1990). Comparative analysis was chosen as the appropriate analysing strategy for the interviews. With such approach, answers from different respondents are grouped together for shared issues (Patton, 1990). In the content analysis, NVivo qualitative content analysis software was used to assist in organizing the data. The analysis started by creating the five predetermined interview themes and theoretical framework into top level codes. Then, each interview was processed from start to end so that the content was coded according to the themes. The data related to a specific theme was codified so that the content was assigned a second level code with descriptive name under the respective top-level code. Where necessary, up to four levels of coding was used to categorize the data and its context. If the data was related to several codes within the theme or in other themes, the relevant data was codified also under those codes. Special care was also taken to any potential and interesting quotations, which were coded to a special "quotations" code. This process was repeated for each interview.

After the content in all interviews was coded in this manner, another pass of the collected data was done from the code hierarchy viewpoint. The content within each code was rigorously analysed so that codes were added, adjusted, and cross-checked as necessary to ensure that all content was codified correctly and comprehensively. The code structure in both phases was created through inductive analysis, forming the codes based on respondent's data both via indigenous concepts and sensitizing concepts. According to Patton (1990), indigenous concepts emerge from the responses and words articulated by the re-

spondents, while sensitizing concepts are created by the researcher by giving names and descriptions to issues indirectly meant by the respondents.

Finally, the resulting code structure was revised to ensure that different codes were on the correct level, and they reflected the proper hierarchy and relationships of the codes. The resulting analysis contained different codes accompanied with the occurrence of that code in the interviews, and the number of interviews the specific code occurred.

5.3.3 Combined analysis

In qualitative research, the analysis can be done in linear fashion, but in quantitative research the analysis is often more circular and developmental (Hirsjärvi et al. 2009). In this study preliminary analysis of the data collected for the quantitative survey was done first. Then, the analysis of the qualitative interviews was done as described in the previous section. During the analysis of the qualitative interviews, the analysis of the quantitative survey data was continuing in parallel. The survey also produced a small portion of textual answers in open-ended questions, which were added to and analysed as part of the interview data in qualitative manner. Finally, both were analysed together in order to arrive to the inferences.

During the combined analysis, a good descriptive overview about the ISPMA's SPM Framework practitioners and current state of process implementation was built. Detailed information about best practices in software product management process improvement and benefits gained by doing so, was formulated into an understandable and logical structure. After the collected data was organized and analysed, the next part was to report the integrated results.

Detailed analysis and implications of the material and results are reported in chapter 6. Results, however, are not complete without researcher's interpretation of the findings. By interpretation, it is meant that the researcher discusses the results of the analysis and makes own conclusions about them (Hirsjärvi et al., 2009). The final chapter of this thesis discusses the results in comparison to prior research.

5.4 Reliability and Validity

In every research, there is an attempt to avoid errors. However, in practice the reliability and validity vary, for which reason they must be evaluated as part of the research efforts. Reliability means the repeatability of measurement or research results – its ability to provide non-random results. Validity on the other hand, means the ability of the measurement or research, to measure exactly what it should. (Hirsjärvi et al., 2009)

In mixed methods studies, Venkatesh et al. (2013) have suggested that it makes sense to differentiate the quality terminology from traditional quantita-

tive and qualitative studies. They have further defined the following new terms which correspond to those used in single-method studies: *inference quality* refers to validity, and *data quality* refers to reliability in mixed methods research. Inference quality consists of *design quality* (i.e., whether a mixed methods study adheres to commonly accepted best practices) and *explanation quality* (i.e., standards for the evaluation of accuracy or authenticity of the conclusion). Data quality is a precondition for validity of quantitative research. However, in qualitative research, one can argue that demonstrating validity is sufficient to establish reliability on its own.

In this study an integrative framework from Venkatesh et al. (2013) is used to assess the quality of the study. The framework provides definitions and examples of quality criteria for mixed methods, which are split into two quality aspects: design quality and explanation quality. These aspects, which also cover the traditional discussion on reliability and validity, are detailed in the following subsections.

5.4.1 Design quality

Design quality means "the degree to which a researcher has selected the most appropriate procedures for answering the research questions" (Venkatesh et al., 2013, pp. 44). Design quality aspects are evaluated against the following criteria from the framework: design suitability, design adequacy and analytic adequacy.

Design suitability covers if the research design and methods are selected so that they are suitable answering the research questions (Venkatesh et al., 2013). Understanding the purposes for using mixed methods is a precondition for suitable research design (Venkatesh et al., 2013). During research design, utilizing only single methods was first considered. The research could have been made as a pure quantitative survey, but it was rejected due to concerns: the response rate was expected to be too small to be representative enough, and as the full list of members was not available, it would not have been feasible to implement a sampling method suitable for a pure quantitative research. Another considered approach was to do the study purely with quantitative interviews, but since the global software industry is very large, an overview of transferable best practices and benefits would not be feasible to create in a master's thesis by one researcher in the limited timeframe. If that approach would have been selected, then the current scope for the international ISPMA's SPM Framework would not have been possible study, and serious scoping would have to be done for example based on specific companies, industries, or countries. Therefore, the design of the research turned into mixed methods, where suitable approach was finally found. The reasoning for the selected research design, a sequential mixed methods study utilizing quantitative survey and qualitative semi-structured interview methods, was described in-depth in section 5.1.

Concerning design adequacy, one must consider whether the research design components are implemented with acceptable quality and rigor (Venkatesh et al., 2013). For quantitative research, indicators of quality include reliability

and internal validity (Venkatesh et al., 2013). In the quantitative survey, its reliability was ensured by defining clear questions and response options and validating those with pre-testing. Internal validity was not specifically sought for, since there the study did not study causal relationships between studied variables. For qualitative research, indicators of quality for design adequacy are credibility and dependability (Venkatesh et al., 2013). Credibility of the interviews was ensured by preparing the outline of the semi-structured interviews with well-thought, formulated questions. It was also validated in the test interview, although having multiple test interviews could have provided even further confirmation. Dependability of the qualitative interviews was ensured with rigorous description of the research process and sticking with the selected process throughout the qualitative inquiry.

In analytic adequacy, the focus is on whether the data analysis procedures are appropriate and adequate to get plausible answers to the research questions (Venkatesh et al., 2013). Indicator for quality for quantitative research is statistical conclusion validity and for qualitative research, the indicators are theoretical validity and plausibility (Venkatesh et al., 2013). The methods for analysing the quantitative survey data were simple, and only useful for limited descriptive purposes. However, as the sample size was also too small for accurate conclusions, more analytical methods would have not provided much additional value either. Therefore, the statistical conclusion validity of the survey data is relatively weak on its own. Concerning the qualitative interviews, theoretical validity was ensured by thoroughly describing the analysis process including the methods of organizing collected data into themes and further into codes, in context of the theories related to the interview questions. Plausibility of the study was ensured by including quotations from the interviewees in the results in abundance. However, there is always a possibility for misinterpretation or miscoding the textual representations of the interview data into codes. The mitigation strategy for this was re-coding and analysis of interview data from multiple viewpoints, and in multiple passes, during the analysis.

5.4.2 Explanation quality

Explanation quality is defined by Venkatesh et al. (2013, pp. 44) as "the degree to which credible interpretations have been made on the basis of obtained results". It is evaluated against the following criteria from the framework: quantitative inferences, qualitative inferences, and integrative inferences.

Evaluation of quantitative inferences considers the degree of accuracy with findings, consistency with theory and generalizability and the indicators for quality are internal validity, statistical conclusion validity and external validity (Venkatesh et al., 2013). As stated in the previous section, the internal validity was not specifically pursued, and the statistical methods used for analysis were very simple. Concerning external validity, the results of the quantitative research are not generalizable to a larger population due to the following reasons: the sample may be biased due to voluntary response data collection

method, and the realized sample size was too small for generalizable results. Although the descriptive statistics and explanations from the quantitative research were not generalizable on their own, they provided both interesting backdrop for the qualitative research, and insight to formulation of the qualitative interview questions.

Qualitative inference considers similarly degree of accuracy with findings, and consistency with theory, but focuses otherwise on transferability instead of generalizability (Venkatesh et al., 2013). The indicators for quality are credibility, confirmability, and transferability (Venkatesh et al., 2013). The credibility of the explanations in the qualitative research builds on the rigorously performed design, and analysis of the interview data as explained in the previous sections. The research process has been described in such detail for the purpose of ensuring the credibility of the research, the results, and implications for theory and practice. Similarly, in the reporting of the results, significant amount of direct quotations from the interviewees have been used to increase transparency and strengthen the confirmability of the results and the research process. The fact that the in-depth qualitative study utilized interviewees from several different organisations, increases the level of transferability of the findings to other contexts. The summarized findings of the qualitative research are mostly aligned with theory and state of knowledge.

Finally, integrative inferences – or meta-inferences – are evaluated from three sub criteria: efficacy, transferability, and correspondence (Venkatesh et al., 2013). The efficacy considers the degree of integrating inferences from separate research methods into a meta-inference, transferability looks at the generalizability and transferability of the meta-inferences, and the correspondence reflects to the initial purposes for using the mixed methods in the first place (Venkatesh et al., 2013). Concerning the efficacy, the qualitative and quantitative research were integrated in two phases: firstly, the quantitative study fed insights into the qualitative study, and secondly the final analysis of the results was a combined effort, resulting to rich and descriptive results. Due to this linkage, the different strands of research were in fact necessary and inseparable in producing quality explanations within the research context.

The transferability of the meta-inferences is partially a result of utilizing two different methods, as it increased the completeness of the picture of the researched phenomena in different settings. Although those were not studied in detail, and every organisational context is unique, the findings and explanations provided by the research have good basis for transferability to different contexts and settings within the users of the ISPMA's SPM Framework.

Concerning integrative correspondence of the research, the mixed method approach was the initial research strategy, and both research methods supported the same research goals in effort to answer the same research questions. The different methods were utilized for complementary, completeness, developmental, and compensation purposes. Although in the end the quantitative survey had primarily descriptive value concerning the context, it provided also important input to the qualitative research and to the combined findings.

6 RESULTS

The previous section described how the study was planned and executed. This section continues further into the topic by reporting the results of the research. In the first section, a short overview of the survey research participants is described via demographic information. The second section describes the interview participants further, which are a subset of the survey participants. The third section describes findings regarding software product management process maturity, reasons for improvements, and challenges. The fourth section describes success factors for process implementation based on the ISPMA's SPM Framework, and the last section describes benefits for improving processes based on the framework.

In reporting of the findings in this section, quantitative words such as "all", "majority", and "many" are used in addition to exact numbers of respondents or percentages. These words are meant to give the reader an overview of the occurrence rate of the phenomena but should not be used for statistical conclusions. Quotes from interviewees in the results have been improved for readability by removing repeated words, spoken language fillers and anonymized data such as people and company names. In some cases, a clarification of context for the quote is clarified with square brackets within the quote. Additionally, removed text blocks in the quote are indicated by square brackets with three dots inside.

6.1 Survey participants

The invitations for the survey were sent to the ISPMA's email distribution list as described in chapter 5. At the end of the survey phase, 26 individual respondents participated to the survey. The basic demographic information about the participants is shown below (figure 11).

n=26

Age	Respondents	
25-34 years	3	12 %
35-44 years	15	58 %
45-54 years	7	27 %
55-64 years	1	4 %

Gender	Respondents	
Male	21	81 %
Female	5	19 %

Country	Respondents	
Germany	6	23 %
Sweden	5	19 %
Netherlands	4	15 %
Switzerland	3	12 %
Finland	2	8 %
France	1	4 %
Portugal	1	4 %
Ireland	1	4 %
India	1	4 %
Denmark	1	4 %
Austria	1	4 %

Job title	Respondents	
Product Manager	14	54 %
Head of Product Management	4	15 %
Product Owner	2	8 %
Project Manager	1	4 %
Program Manager	1	4 %
Others: Industry Principal, Portfolio Manager, Head of Products, Software Project and Product Management)	4	15 %

FIGURE 11 Age, Gender, Country, and Job title of the survey respondents

The age of the respondents was mostly (85%) between 35-54 years, and 81% of the respondents were male. The respondents in the study come from 11 different countries, which of only one was from outside Europe. The most used job title for the respondents was "Product Manager" (54%), with "Head of Product Management" as the second most common response (15%). In addition, the respondents gave several other job titles outside of the pre-defined answering options.

The respondents were also asked to provide information about their software product management certifications, especially related to the ISPMA. The results are shown below (figure 12).

n=26

ISPMA SPM certificates held	Respondents	
Foundation Level	20	77 %
Foundation Level and Excellence level(s): Product Planning	2	8 %
Foundation Level and Excellence level(s): Product Strategy, Strategic Management	2	8 %
Foundation Level and Excellence level(s): Product Strategy	1	4 %
Foundation Level and Excellence level(s): Product Strategy, Product Planning, Strategic Management	1	4 %

Colleagues with ISPMA certificates	Respondents	
0	5	19 %
1-5	10	38 %
6 or more	7	27 %
Don't know	4	15 %

Other certificates	Respondents
Pragmatic Marketing Foundation	3
CSPO	2
PSPO	1
AIPMM	1
Product Focus	1
Open Product Management Workflow	1

FIGURE 12 Certifications of the survey respondents and their colleagues

All respondents have at least ISPMA SPM Foundation level certification, and 24% of the respondents have also at least one ISPMA Excellence level certification. One third (35%) of the respondents have also various other certifications related to Software Product Management. 65% of the respondents also had colleagues in their company with ISPMA certifications. Most commonly, other certifications were related to the Pragmatic Marketing framework or different variants of Scrum Product Owner certifications such as CSPO and PSPO.

The respondents were also asked to provide information about the organisation they work in to get further background demographics. The results are shown below (figure 13).

n=26

Organisation	Respondents	
Product organisation	17	65 %
Corporate IT organisation	5	19 %
Other	4	15 %

Size of the organisation	Respondents	
Micro enterprise (less than 10 employees)	0	0 %
Small enterprise (10-49 employees)	2	8 %
Medium-sized enterprise (50-249 employees)	3	12 %
Large enterprise (250 or more employees)	21	81 %

Industry of the organisation	Respondents	
Software products and software-intensive technical services (software industry)	19	73 %
Software-intensive products (other industries)	6	23 %
Professional human services (other industries)	1	4 %

Sector of the organisation's customers	Respondents	
Profit	17	65 %
Public	8	31 %
Non-profit	1	4 %
None of the above (Individuals)	0	0 %

FIGURE 13 Survey respondent's organisations

73% of the respondents identified themselves working in a Product Organisation, and the others were working with Corporate IT Organisations (19%), or other type of organisations. 81% of the respondents are working in large enterprises (250 or more employees), and there were no respondents working in micro enterprises (less than 10 employees).

73% of the respondents classified their company in being within software industry (software products and software-intensive technical services), and 23% indicated other industries with software-intensive products. 65% of the enterprises were providing products and services for the profit sector, and 35% for the public or non-profit sectors. None of the respondents stated that their companies' primary customers would be private individuals.

To get more information about the companies, the respondents were also asked to provide information about the product management and the products of the company. The results are shown below (figure 14).

n=26

Number of software products	Respondents	
0	1	4 %
1	2	8 %
2-5	4	15 %
6-20	10	38 %
20-99	5	19 %
100 or more	3	12 %
Don't know	1	4 %

Type of products	Respondents	
B2C products for individuals	1	4 %
B2B products for small organizations	0	0 %
B2B products for medium organizations	1	4 %
B2B products for large organizations	5	19 %
B2B products to varying sizes of organisations	14	54 %
A mix of B2B and B2C products	5	19 %

Typical product lifetime	Respondents	
0-1 years	1	4 %
2-3 years	0	0 %
4-5 years	2	8 %
6-9 years	6	23 %
10 years or more	17	65 %

	Product Management		*All of Product team*	
Size of organisation	Respondents		Respondents	
0 persons	0	0 %	0	0 %
1 persons	0	0 %	0	0 %
2-10 persons	13	50 %	1	4 %
10-99 persons	8	31 %	11	42 %
100 or more persons	5	19 %	14	54 %

FIGURE 14 Product management and products in the survey respondent's organisations

Half of the respondents reported that their product management function consisted of 2-10 persons. Over half (54% of the respondents) reported the size of

the whole product function, including all employees working on the products, to be over 100 persons.

Majority (54%) of the respondents categorized their companies' products as B2B products for varying sizes of organizations. Only one respondent categorized their company's products as B2C, and one fifth (19%) of the respondents reported having a mix of B2B and B2C products. The majority, almost 40% of the respondents reported their company having from 6 to 20 products, but a few had over 100 different products. A vast majority (65%) of the respondents stated that the typical lifetime of their companies' products is 10 years or more.

Finally, the respondents were asked to provide information about process improvement based on the ISPMA's SPM Framework in their organisation. The results are shown below (figure 15).

n=26

Process implementation based on ISPMA's SPM Framework	Respondents	
Yes	4	15 %
Partially	20	77 %
No	2	8 %

Involvement in the process improvement	Respondents	
Participated to the process improvement	15	58 %
Led the process improvement	4	15 %
Not involved	7	27 %

Last time of process improvement	Respondents	
Within last 3 months	3	12 %
Within last 6 months	4	15 %
Within last 1 year	2	8 %
Within last 3 years	4	15 %
Within last 5 years	1	4 %
The processes are continuosly being improved	6	23 %
Never	6	23 %

Other frameworks used to improve processes	Respondents
Scaled Agile Framework (SAFE)	9
Pragmatic Marketing	5
Product Focus	1
Open Product Mgmt Workflow	1

FIGURE 15 Implementation of the ISPMA's SPM Framework in the survey respondent's organisations

Practices based on the ISPMA's SPM Framework have been implemented in the organisations of 92% of the respondents, and 73% of the respondents have been either leading or participating to the process work. 35% responded that such implementation or improvement work has occurred during the last year, and 23% of the respondents stated that processes are improved continuously and systematically. In addition to the utilization of the ISPMA's SPM Framework, 58% of the respondents stated that they have utilized also other frameworks to improve software product management processes. Scaled Agile Framework (9 respondents, 35% of all respondents) and Pragmatic Marketing (5 respondents, 19% of all respondents), were the most utilized frameworks.

6.2 Interview participants

Since all the interviewees also participated in the survey, the survey data could be mapped with the more in-depth information received during the interview to get a better overview of the interviewees and their companies. The table below (table 11) shows an anonymized summary of the companies the interview subjects were employed in, when combined with the survey data.

TABLE 11 Interviewees and companies they are employed in

#	Title(s)	Type of business	Size of organisation	Software products
1	Portfolio Manager	Software products for B2B customers. Organized as a product organisation.	Total employees: 18 000 Product Management: 10-99 Product organisation: >100	20-99 products Typical lifetime 10+ years
2, 8	Head of Product Management, Portfolio Manager	Software-intensive products for B2B customers. Organized as a product organisation.	Total employees: 700 Product Management: 2-10 Product organisation: >100	6-20 products Typical lifetime 10+ years
3	Head of Product Management	Software-intensive products for B2B and B2C customers. Organized as a corporate IT organisation.	Total employees: 65 000 Product Management: >100 Product organisation: >100	>100 products Typical lifetime 10+ years
4	Head of Product Management	Software products for B2B customers. Organized as a product organisation	Total employees: 1 700 Product Management: 10-99 Product organisation: >100	6-20 products Typical lifetime 6-9 years
5	Product Owner	Software products for B2B customers. Organized as a product organisation	Total employees: 200 Product Management: 2-10 Product organisation: 10-99	1 product Typical lifetime 10+ years
6	Product Manager	Software products for B2B and B2C customers. Organized as a product organisation	Total employees: 70 Product Management: 2-10 Product organisation: 10-99	2-5 products Typical lifetime 10+ years
7	Product Manager	Software products for B2B customers. Organized as a product organisation	Total employees: 3 000 Product Management: 2-10 Product organisation: 10-99	6-20 products Typical lifetime 10+ years

All the interviewees were from separate organisations, except for respondent 2 and 8, which were from the same company. The interviewees were a mix of Product Managers, Portfolio Managers (having responsibility over multiple products), and Head of Product Managers. Majority of the companies were large enterprises (250 employees or more), two of which have over 10 000 employees. Most companies where the interviewees are employed in, serve only B2B customers, but in two cases the companies had also consumer products (B2C). For 50% of the companies included, the size of the product management organisation was between 2-10 persons, while the other companies were gener-

ally larger, and had also larger product management and overall product organisations.

The interviewees employed by the two largest companies reported that they have tens, or even hundreds of different software products. Other companies had typically 6-20 different software products, except for one interviewee stated that in their company there is only one software product to manage. All except one interviewee indicated that in their company the typical lifetime of their products is 10 or more years, from inception to end-of-life.

All interviewees reported that they have been involved in software product management process implementation or improvement work in their company. Most of interviewees said that the process has started during the last three years, and Respondent4 had the most recent experiences since the process has started only eight months ago. Two of the interviewees from the two largest companies, said that they started the work already five years ago.

Furthermore, all interviewees stated that the process improvement in their company has been fully based or at least influenced by the ISPMA's SPM Framework. In five out of seven companies where the interviewees work at, the software management processes are used with all software products. In the rest of the companies, the processes are in use only for some of the products with plans to either expand the use or take an alternative approach. During the research, participants were asked to evaluate maturity of their software product management processes, what are the reasons for process improvement, and how they are organized around the improvements. These results are described in this section first based on the survey research, and then based on the findings in the interviews.

6.2.1 Overall process maturity

In the quantitative survey, respondents were asked to evaluate how mature they are saw their company being in the different areas of the ISPMA's SPM Framework. The results of the survey are shown below (figure 16).

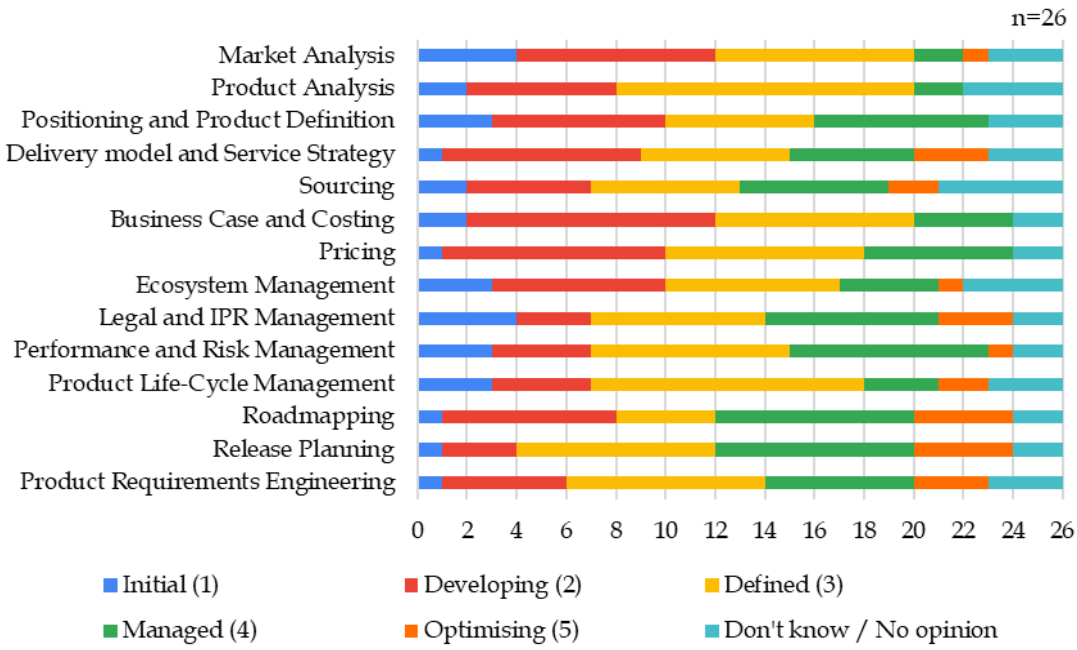

FIGURE 16 Process maturity in the respondent's companies (current status)

The respondents were asked to subjectively estimate their process maturity in the following scale:

- Initial: Processes are unpredictable, poorly controlled, and reactive
- Developing: Processes are defined in project level and are often reactive
- Defined: Processes are defined in company level and are proactive
- Managed: Processes are measured and controlled
- Optimising: Focus is on process improvement

CMMI maturity levels were used as a model for the levels and their short definitions for the survey. For the purposes of presenting the results, the levels were given a value from 1 ("Initial", the least mature level) to 5 ("Optimising", the most mature level).

In addition to evaluation of the current state, the respondents were also asked to select the five most important process areas, what they thought their company should improve next. The results are shown below (figure 17).

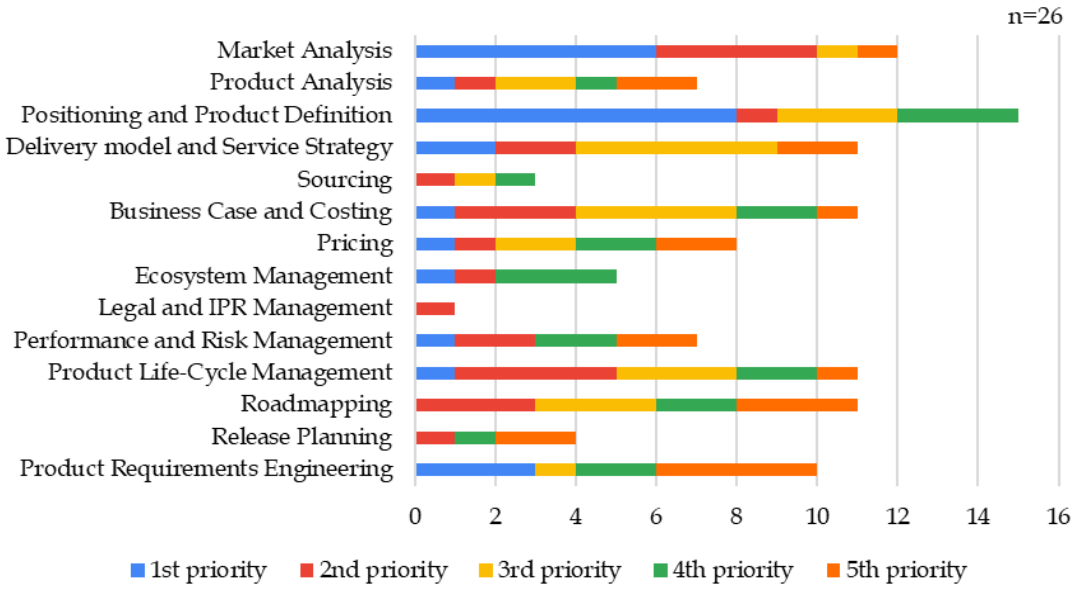

FIGURE 17 Five most important process focus areas in the future

By calculating weighted arithmetical mean on the estimated maturity levels per process area, a ranked order of maturity was established. This is shown below (table 12), combined with information on what the respondents stated the next focus areas should be.

TABLE 12 Order of maturity for ISPMA's SPM Framework core processes

Process Area	**Maturity level**	**Focus area selections (total)**	**Focus area selections (1st priority)**
Release Planning	3,46	4	0
Roadmapping	3,29	11	0
Product Requirements Engineering	3,22	10	3
Legal and IPR Management	3,08	1	0
Sourcing	3,05	3	0
Delivery model and Service Strategy	3,04	11	2
Performance and Risk Management	3,00	7	1
Product Life-Cycle Management	2,87	11	1
Pricing	2,79	8	1
Positioning and Product Definition	2,74	15	8
Ecosystem Management	2,68	5	1
Product Analysis	2,64	7	1
Business Case and Costing	2,58	11	1
Market Analysis	2,48	12	6

The column "Maturity level" contains weighted arithmetical mean of the reported maturity levels per process area. The weighting is done related to amount of responses per process area after removing "No opinion" responses. A value of "2" in the column corresponds to "Developing" maturity level and

"3" corresponds to "Defined" maturity level. However, it should be noted that the scale is more of an ordinal scale, and not a fixed difference interval scale.

The column "Focus area selections (total)" contains the number of how many times in total the process area was selected within the five most important process areas to focus on. Respectively, the column "Focus area selections (1st priority)" contains occurrences of how many times the process area was selected as the most important area (1st priority) to focus on.

The most mature areas were found to be "Release Planning" and "Roadmapping", as 46% of the respondents stated that those processes were already on the "Managed (4)" or "Optimising (5)" level. Those areas also have the highest overall maturity level based on the weighted averages. The process areas related to "Product Analysis", "Positioning and Product Definition", "Business Case and Costing" and "Pricing" had not reached "Optimising (5)" level in any of the responses. All these areas also belonged to the 5 least mature areas based on the weighted averages.

Two of the process areas were selected the most as 1st priority to focus on next: "Positioning and Product Definition" and "Market Analysis". The same two areas are also the most important when sum of all priorities is calculated. When looking further into the sums, the least important areas to improve on next based on the survey were "Legal and IPR Management", "Sourcing" and "Release Planning".

When asked about the reasoning why the respondent selected those areas the most important improvements areas, the reasons were very specific for the situation of the company and the products.

6.2.2 Assessment practices

In the qualitative interviews, it was further studied how the assessment of the process maturity has been done in different companies when process improvements have been going on. The fundamental of assessing the current situation was according to the most interviewees discussions and observation of the existing ways of working and the processes or lack of those.

> We have gone and looked at what we do. We have looked at the processes within individual teams such as development teams. We have gone around with some of the other parts of the business that we have to work with as well. You know, customer support, sales to see what if they had any processes in place, so that we could make sure that everything we introduced was aligned with how they work in so far as possible. And otherwise really, it was just kind of through observing and through living through some really difficult challenges with some of the products. (Respondent3)

Another information source for the assessment was mentioned by some the documentation about the process, or the artifacts provided by the processes. For the processes, the process documentation was non-existent according to some of the respondents, and had to be created during the assessment as stated by

Respondent1: “I tried to basically document our existing processes and then compare that with those ISPMA framework”.

The artifacts related to the processes, were more easily found during the assessment. “I just checked through the JIRA projects, JIRA tickets, Confluence pages, documentation and through PowerPoints” (Respondent4). However, in many cases the artifacts were related only to the development related process bits of the software product management:

> We have quite a stable set of tools that are used again within the development teams for their day to day work, work item tracking, release, planning, all those things. But when it comes to product management, we haven't adopted any product management tools. (Respondent2)

The existing tools which were used in the software product processes were used by some of the respondents to make examples on how the new and old way of working would change, and how the processes would be supported by those tools.

> We had some tools used in product management, so we tried to use the functionality to kind of assess or see what we could implement. […] It was just basically using our existing tools to showcase the differences in some way. (Respondent1)

For companies which had been practicing software product management in a formalized way for a longer time, the existing process metrics were also a way assess the maturity, as stated by Respondent3:

> We try to be always data driven. So in a lot of the cases when it comes to quantitative, like, you know, applying the standards, or making sure that we all have product strategies signed, approved and in place, uploaded and visible to everyone, those things were measured. And so it's part of our business-as-usual process of SPM to show the statistics around that. (Respondent3)

Some found the assessment to be more straightforward – there was no product management processes to start with, as this was the first implementation. As some of the respondents put it: “The assessment was that there is no product management” (Respondent7) and “I didn't really assess it because we didn't have any process before” (Respondent8).

One respondent had first-hand experience in implementing software management processes both in a small company, and a larger company. He found the assessment easier to do in the small organisation:

> [In the smaller company] it was easier to do because I knew our processes, it was much simpler processes, smaller groups when trying to compare that to the ISPMA framework, how we are organized and so on. So that was not too difficult in some sense to do because it was fairly flat hierarchy as well and I basically knew all the people and all the processes involved. In the larger organization where we are now, there it is more difficult. (Respondent1)

One of the respondents also mentioned that they have utilized an SPM Maturity Matrix (Bekkers et al., 2010), and been part of a research concerning software product management process maturity a long time ago, to assess the level of their software product management processes.

> What we did, years ago, we also used a tool to measure the level of the product management process. […] Of course it could help you. Like always with such a tool, it's dependent on what you put in there. […] It was useful just to see, to do it once in a while to see we were standing, and if there are things where you could improve. (Respondent5)

Respondents were also asked if they could think of practices or tools which would have helped them in the maturity assessment. Two themes emerged: checklists and best practice maturity levels. According to Respondent4, a summary of the ISPMA's SPM Framework and checklists would help getting started in a new process improvement situation:

> A checklist would have been useful […] when you join a new company, you don't just have all of that information in a nice linear or chronological checklist of what to go through in your mind. (Respondent4)

One of the respondents also proposed defined maturity levels specifically for the framework to assist with planning of the process improvements:

> I think it might be helpful have for every component of the ISPMA model best practice levels that you can achieve as an organization. And if they are defined from more theoretical perspective, for instance, by some kind of meta-study over multiple firms or companies, just to see how successful companies have organized their software product management tasks, that might be helpful. That can show you what are the next steps that are ahead of you, if you want to go further. And what do you have to be thinking about right now, and what do you have to worry about in a few years. (Respondent6)

6.2.3 Reasons for process improvement

In roughly half of the interviews, the process implementation or improvement work was said to have been initiated by the product managers, or other experts working with the products. The other half was initiated by the company management, in many cases after discussions with external consultants. The reasons why the framework adoption and process improvements were initiated, varied between the interviewees.

The most common driver for software product management process improvement was the lack of market orientation in the existing ways of working. Several interviewees stated that their focus was too much on the technical side of the products, instead of looking more at the customer, market, and competitive landscape. For example, Respondent5 stated that "we are now very focused

on the product and very focused on details" and "we want to position them to be more market oriented".

On a positive note, in some interviews the technology choices and development methods used with the products were mentioned to enable building new features fast – but the lack of managing what to build, caused issues and was the reason for process improvement:

> But we were lacking the business or strategic or market insights to say what we should be building. It was very developer led. Here, we can build these cool things, but then hang on, who asked for them or are they going to sell? (Respondent2)

The interviewees had issues both in product planning, roadmapping, and lifecycle management, which were drivers for improvement. As Respondent1 put it, "I felt that we were doing very product planning in a very amateur way". Respondent4 had issues specifically related to the requirements management for different markets, as "you have these different voices screaming out for what they need the software to do, and sometimes they're very conflicting". Pleasing different markets and orchestration with sales was also concern for Respondent8:

> You need to be aware of what is the consequences of putting this in, what is... because you might make a sale, but was is the cost, the lifetime cost of actually adding this? (Respondent8)

In one of the interviewees companies, there was also ongoing change of business model, which was the primary reason for the change. The organisation was earlier focused on providing services, and via identifying reusable components of solutions, had started to shift into more standardized software products. This was also seen to require a change in the ways of working.

The interviewees were also asked to detail what are the more concrete triggers for doing the process improvements. The most frequently mentioned trigger was feedback. By feedback the interviewees meant both feedback from the process and staff, but also from the management or other stakeholders. Respondent7 linked the triggers for improvements for regular retrospectives: "Based on different retrospectives, there are some triggers for improving the processes". Similarly, Respondent3 reported a regular way of identifying improvement needs:

> Most of the time the need for improvement is based on the feedback from the stakeholders, but also from the SPM community itself […] which is active and very regular (Respondent3).

Top management was also separately mentioned by several respondents as a major source for triggering the improvements. Respondent1 stated that "It's usually the higher management who initiate changes" and "we have a fairly hierarchical approach". Respondent5 concretized the input from management with a practical example that "We received feedback or input from manage-

ment" which resulted within the product management the goal that "we want to be more market oriented and have more contact with our customers".

Other mentioned triggers were changes in corporate strategy, systematic review of the process performance, exposed shortcomings, and need to improve internal communication.

6.2.4 Organizing around the improvements

It was asked in the interviews how the companies have been organized around the software product management process improvements. According to the interviewees, in most cases it was the Head of Product Management and Product Managers who formed the main group for the improvement work. In organisations where there were also Product Owners, they we also part of the process improvement team. In some cases, also the management board or even the CEO was part of the process improvement work. Depending of the organisation size and operating model, interviewees mentioned also other parts of the organisation which were also involved in the work: sales, customer experience, release train engineers (SAFe), architects, and project management.

Smaller improvements were done according to the interviewees based on the feedback cycle from the process, which was typically 2-3 weeks. Most of the interviewees reported pre-defined cycles and regular improvements, but for some it was more ad-hoc. The feedback from the process via both formal and informal channels was seen crucial in defining the smaller changes to the processes. The interviewees felt these smaller changes to be "continuous" improvements compared to larger improvement work what was often put into a workgroup as exemplified by Respondent3:

> When I joined, we had to rejuvenate the product management because of the Agile and we had a couple points plan which we created as a result of a working group. My boss created six or so small working groups consisting of SPMs itself. And within those working groups they had about maybe four weeks to... First of all, identify pain points, and then formulate the problem statement and then of course try to figure out what would be the necessary change. (Respondent3)

These larger changes to the processes were done in longer intervals, ranging from few months to several years. Respondent1 said that "Everybody has two years to smooth things out and implement these things, and once that's working well, then there's the next big change". Respondent7 mentioned that in the organisation the process implementation started with high interval regular meetings, which then over time was changed to a more relaxed schedule:

> There were weekly meetings with the CPO (Chief Product Officer) regarding, which KPIs we should define, how we can improve strategy process, how can we build up a 24-month rolling plan, how can we maybe also introduce OKRs and so on. After we got more and more clear about how we interact, the frequency moved to two weeks, and after then it was monthly. And there was similar panel on the development side with product owners. And I would say once a month there was a general alignment

> between all the product owners and managers. Within the products, there was daily alignment between the development and product management. (Respondent7)

In the organisation of Respondent6, there was also a systematic approach to planning the changes for the already established software product management processes:

> What we do in our company is that we have every six months an evaluation, and we look at all the components of the ISPMA framework. And for every component, we define where we are now, and what the next step will be for the coming six months to improve on that. And sometimes there are certain parts of the framework that you decide, we're not going to put much energy in that right now because it's not the right moment - and some parts are heavily worked on, because we need to improve on those very quickly. And that method is very nice because we do that with all parts of the organization. (Respondent6)

Majority of the interviewees also thought that ISPMA certifications were very useful for the improvement process. Discussions with colleagues became easier, when also other people had same understanding of what was being done. Respondent7 described having a knowledge of a framework as "you are working towards a goal that everybody has the same picture of" and Respondent3 highlighted the value of common terminology as "it helped immensely to progress just by using the same language and the same understanding of concepts".

A few interviewees mentioned also that having certificates helped in gaining authority around the process changes in the company. According to Respondent1 having a certification was "helpful in being able to convince people why something is important" and Respondent2 stated further:

> It's given us the authority to suggest changes in process internally. And when people ask why, we can refer sometimes back to the framework. And you know, also, I think open people's eyes to the scope of what's involved in product management. (Respondent2)

In addition to internal authority, almost half of the interviewees said that their companies had used help of external consultants before or during the implementation. External consultancy typically occurred on the management level, and was in many cases done by senior members of the ISPMA organisation. One of the main reasons of using consultants was summarized as "It's not always easy to step out of your daily work and have a good objective view on the organization as a whole" (Respondent6).

6.2.5 Challenges

The interviewees were also asked to describe what have been the challenges in implementing the software product management processes. By taking potential challenges into consideration, one could avoid or mitigate those in their own

improvement efforts. The challenges described by the interviewees are summarized in the list below:

- Change resistance and organisational changes
- Stakeholder understanding and buy-in
- Sales orchestration changes
- Adoption in Agile environments

This section describes these challenges on a general level. The most frequently mentioned challenge was change resistance. It manifested in many forms and was in many times associated with difficulties of managing organisational changes. As Respondent1 put it:

> Anything that involves large organizational changes, it's a bit difficult because it involves shuffling people around. And that somehow always seems to be the most disruptive thing. (Respondent1)

It was also mentioned as a challenge to get buy-in to the process improvements from different stakeholders – first and foremost the management and staff, but also in some cases the customers. As commented by Respondent4, the lack of buy-in can in some cases originate from the lack of understanding what software product management is:

> The most challenging thing has been getting the business side to understand and react to what we're doing. […] They're not software people, and they do did not understand what goes into building a software product. […] They just don't know how to contribute. (Respondent4)

Some respondents mentioned the main challenges in the process improvements was to change how the interactions or orchestration of the sales function of the companies works. Respondent2 commented:

> I would say that anything that involves orchestration with sales, anything that involves on agreeing on what we should be putting in our products to address different categories of customers, is the most challenging part to change. (Respondent2)

Several interviewees also mentioned that matching the software product management framework with agile development models has been difficult, especially with defining the roles correctly:

> We had a period where there was confusion within the organization, namely down to the level what a PO is doing and what this PM is doing now, and what are their responsibilities. (Respondent3)

Within the interviewees, there was also some criticism towards the framework itself, of being "not very optimized for the new agile development" (Respond-

ent5). Additionally, some pointed out that doing too many changes at the same time is difficult:

> You should not do, at least in my point of view, is to do the introduction of software product management at the same time while introducing agile development. I would try to keep the change at one time as small as possible. We did lots of changes in the company of the same time, and that led to lots of difficulties in the overall organization process. (Respondent7)

6.3 Best Practices and Success Factors

During the survey research, participants were asked to weigh in on how important they saw different success factors to be concerning process improvement. The results of the survey are shown below (figure 18).

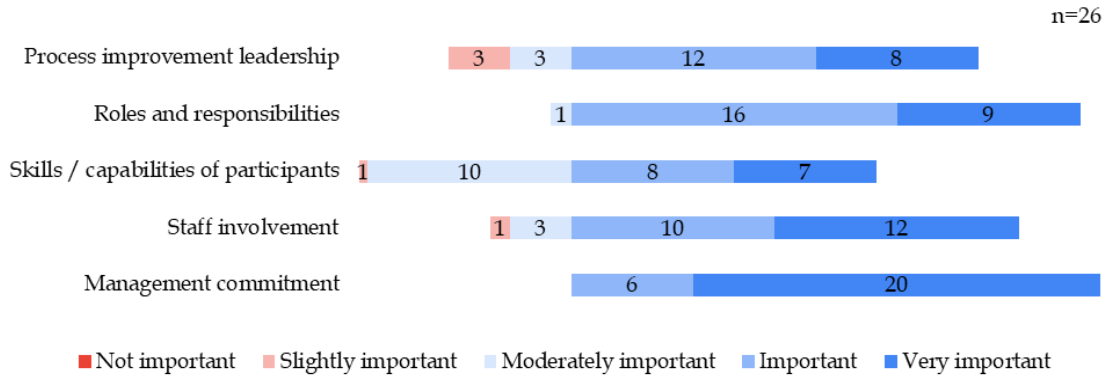

FIGURE 18 Importance of success factors for process improvement

The most important success factor for process improvement was found to be management commitment. 77% of the respondents ranked it on the highest level as "Very Important". The second highest ranking success factor was staff involvement, and third highest success factor by a small margin, was roles and responsibilities. None of the success factors was ranked as "Not important" by any of the respondents.

In the interviews, the three top ranking success factors were under further study, and the respondents were both asked to describe from their view what those success factors mean to them, and how they are ensured in their respective companies, in effort to find out best practices for process improvements. The findings related to these success factors are described in the following subsections.

6.3.1 Management commitment

Management commitment was defined by most of the interviewees as that the management supports the plan or process, which is being implemented, and sticking to that change process despite pushback. The management commitment also involves active participation by the management themselves, trust to the process improvement organisation, consistency in communication and the fact that it is reflected in the funding.

Ways to ensure the management commitment were various according to the interviewees. Within management, there needs to be a high degree of understanding to the process being improved and the goals for the process change.

> That's a problem I've seen at every company I've worked with management, that they are on so high level that they don't understand, sometimes even at the start-up level […] if you want to implement a larger framework like SAFe or ISPMA framework, then you need to get management to buy-in on that, so that they know what they're committing to. (Respondent4)

This so-called buy-in by the management also reaches to the management themselves – they need to actively be part of the process change, and not "just" in the information loop. Respondent6 stated that for example establishing a product portfolio management board including "two of the three directors", convening on a regular basis, creates and reinforces the commitment into the change. Another way to ensure buy-in is to do it by aligning the objectives:

> It is very simple. In my company, KPIs (Key Performance Indicators) and OKRs, Objectives and Key Results. You have to make sure that everybody's KPIs are focused on the same thing or the same path. (Respondent4)

In cases when the improvement initiatives were started by management, it was seen easier to also ensure the management commitment in such top-down approach. As Respondent7 puts it:

> The process started with the CEO. He convinced the management board and it was the task of the management board to convince all the others. (Respondent7)

When the process initiative has been started from the expert level, there is more work to be done with management and staff commitment. As Respondent8 put it, "we have to go to talk to people and, you know, sort of win them over to our way of thinking". Having a good manager as a champion towards the management team helps:

> We are lucky we have a good CTO who I report to, who is very much committed to our success and represents us very strongly as product teams within the leadership team. (Respondent2)

Good and regular communication from the management about the process change and progress in it, both strengthens the management's commitment into the changes, but also the commitment of the staff. The responsibilities regarding the process change need to be communicated as well within management and staff, to ensure there are no unclarities. As Respondent1 put it, "from management side the biggest thing you can do is that they transparently communicate that they are committed to this process".

The consistency is key in supporting the process improvement. It was mentioned by several interviewees that supporting the change initiative by encouraging "following a certain process and avoiding other behaviours" (Respondent2) is crucial in case of pushback from the organisation. Failing to do so by not supporting the initiative under pressure "waters down some of them improving processes that we try to implement" (Respondent1). One of the respondents attributed the whole success of the process improvement to the level of consistency:

> The strength of consistency and commitment of the management is telling me how quick the adoption will be, and how successful it will be. (Respondent3)

In larger organisations, it may also be worthwhile to setup a separate unit for software product management to facilitate the speed and focus of the process improvement. However, according to Respondent3 "like in any adoption, having organizational unit becomes an obstacle in further adoption and integration" as the maturity progresses.

6.3.2 Staff involvement

Staff involvement was defined by the interviewees as that the actual people who would be working with the processes, are involved in the process implementation or improvement. Staff involvement ensures alignment, shared understanding and focus concerning the process improvement. Feedback cycle between the management and staff concerning the changes is a crucial part of the staff involvement.

There were several different best practices found, which positively contribute to staff involvement during the process improvements. According to most interviews, having a good degree of freedom within the staff to define and refine the process, is a major contributing factor to the staff involvement. One example of how to manage this, is to establish and maintain an expert-led process change group. In the organisation where Respondent3 works in, these so-called SMEs (Subject Matter Experts) are also the drivers of an established "community of practice" within the company for all things considering software product management.

> It's a community that is active and very regular... it has a role of giving guidance and keeping up to date practices, and also receiving feedback. […] [It consists of] the

> software product managers, people that that are certified with ISPMA and people that practice or used to practice that. (Respondent3)

As with the management commitment, having a cadence of regular meetings and communication is also one of the key findings for success. Regular meetings "at all levels in the staff" (Respondent1) and "making everybody aware of the fact we are doing a transition" (Respondent6) is important.

There needs to be open dialogue possibilities about the process improvements and the management should be "straightforward in asking what could be done better, are there any other problems and gathering general feedback" (Respondent1). One approach to facilitate feedback was mentioned by Respondent3 to be the beforementioned expert-led community of practice:

> Because the group, the community of practice, is really a community run by SMEs, it's not "management" ... there is a little bit more perhaps openness to provide a feedback as well. (Respondent3)

In general, too strict management dictated top-down approach may not yield desirable results:

> It's not like that the management is saying "you are now going to do this and this or that way", because I don't think that it will be a success. So now we are more free and we can do it our own way. (Respondent5)

Keeping the staff involvement high from the start, can be easier in cases where the process initiative has been started from the expert level, than in initiatives "run by management":

> The ISPMA adoption always has and continues to be pretty much bottom up initiative [by the SMEs]. And with that, the management never had to invest too much time to get the staff involved, because product management practice was never considered a top management directive. It was always coming from the bottom of the organization, from the people that actually know and do manage software on a daily basis. (Respondent3)

As a management tool, KPIs were mentioned by Respondent4 as useful for ensuring staff involvement. When the KPIs for the staff are aligned appropriately with the process improvement goals, they support the direction and ensure focus.

According to Respondent8, "Everybody wants to do things right. And a lot of them have been quite frustrated with the way we have been doing things earlier". Being in a bad situation earlier, and seeing the process improvements in practice working is a contributor to further staff involvement:

> I do think that my team is really, really buying into this, because the changes we've done over the last year, the amount of optimizations we've managed to implement, actually, everybody can see this works […] I really think that the results are speaking very loudly for themselves. (Respondent8)

Additionally, the staff involvement in the improvement process should not be limited just to the core staff in the process – instead, persons should be included quite extensively:

> We also try to involve people from support, from sales, even people from service delivery […] we recognize that we need their input because they spend an awful lot more time in contact with customers and prospects than we do in our product teams. (Respondent2)

In fact, a practical problem may be the sheer about of people who should be involved, as the whole range of software product management activities and processes needed can be very pervasive.

> The problem we have had is that where do you stop and who do you involve, because we have tried many different approaches […] If we wanted to get a really deep view, we have to bring about ten or fifteen people from every business unit and we would be completely overwhelmed with conflicting input. (Respondent2)

6.3.3 Roles and responsibilities

Roles and responsibilities for the process improvement, was defined by interviewees as the clearly communicated roles of the process participants and what specific tasks they are expected to. Everyone cannot be working with everything, and there needs to be clear responsibilities and decision making. To succeed in the process improvement and the resulting processes, the roles must be also resourced accordingly.

According to the interviewees, making the roles and responsibilities clear is a challenging task. Even just ensuring that clarity is understood in the same way can be challenge:

> I think one thing that we overestimated, is our understanding of what does it mean to provide clarity around the roles and responsibilities, and what does staff actually consider as clear or at what stage. (Respondent3)

The roles need to be clear both during the process improvement, and in the resulting processes. As respondent1 put it:

> Everybody should be somewhat involved in that, but it means that not everybody can do the same thing […] Not everybody can be responsible for the improvement process, making decisions about that. But they must be responsible in giving correct feedback about the improvements and about the process. (Respondent1)

Involving a lot of people to the improvement process and discussing different details in-depth was the key of finding the right roles. A few respondents mentioned using the ISPMA's SPM Framework structure directly as a basis for these discussions both with staff and management. Almost half of the interviewees stated that they have utilized a custom-made RACI or RASCI model in estab-

lishing the proper roles and responsibilities. RACI stands for "Responsible", "Accountable" "Consulted" and "Informed", and charting responsibilities to a such matrix helps in creating a clarified view of the activities and roles within a project or process (Smith, Erwin & Diaferio, 2005). RASCI is one extension of the model, having also separate "Support" role for assisting in completion of an activity (Cabanillas, Resinas, Ruiz-Cortés, 2011). Utilizing such model not only makes the roles and responsibilities clearer, but also helps in identifying gaps which need to be filled:

> We have implemented the RACI model. We have got a very clear table of who is responsible, accountable, consulted and informed at each step of the process. And that has really helped us a lot to go through the development process. […] It is just about ensuring that you have the right roles and responsibilities in place. Right now, I am hiring three people based on the roles we have identified that we do not have. (Respondent4)

In case of Respondent6, their company was implementing the software product management framework at the same time as changing their business model from professional services to standardized products. This caused a lot of changes in the processes and what was expected from the personnel, and the RACI matrix was a good tool to illustrate this change:

> I think stating the fact that roles are changing and defining and making strict responsibilities, helps us to understand what was changing. So it was very important for us and we also made a RACI model, which helped us a lot to look at the old situation versus the new situation. (Respondent6)

According to the interviewees, the RACI model is especially useful when setting up and discussing the processes, roles, and responsibilities, but less important in the daily work:

> I don't think anyone refers to it on a daily basis. It's more just like, OK, we know that guy or that role should really own this part. […] we can go back to the RACI model and say, hey, come on, you needed to own this, but that has not happened yet. (Respondent4)

One of the respondents concluded that a good indication for reaching clearly defined roles and responsibilities is that "When the comments about roles and responsibilities disappear from the feedback, that means that the job is done" (Respondent3).

6.4 Benefits

In the interviews, the respondents were asked to describe what kind of benefits have been gained by implementing or improving software product management processes based on the ISPMA's SPM Framework in their organisations.

This section describes the found main benefits, and in the last subsection, a few other benefits mentioned by the interviewees are also listed.

6.4.1 Increased awareness of software product management

The most often mentioned benefit was an increase in the general awareness of the software product management in the companies. The increased awareness makes things easier with the product team, management, and other stakeholders, as exemplified by Respondent2:

> It has opened people's eyes to the scope of what's involved in product management. We've had many instances of people confusing product management with product development […] It just helps people realize the that you have to be much broader in thinking about all the things you have to do to make a product successful. So it's really helped us with that understanding. (Respondent2).

Having a structure and awareness of the different activities helps with the general alignment in the company in several ways, as described by Respondent1:

> We're able to align people's understanding across the company much better as a result of [the processes making everything much more deliberate and planned]. (Respondent1)

Respondent7 said that the clearer roles and responsibilities give the product managers authority, as "It's much easier to orchestrate things as it's clear that I am, for example, the responsible person for the cost of the product". This is not only towards product teams, but also a better position to challenge the management to drive more engagement as pointed out by Respondent2: "We are able to challenge our senior leaders, for example, to be more deliberate about setting out the product strategy".

However, everyone does not need to agree on all things - and should not, "In a start-up, everyone has an idea about the product" (Respondent5), but the raised awareness helps in pushing through the necessary decisions as described by Respondent4:

> And even if someone might not agree that that's the highest priority, they still understand what will be built and why. So, I think that that just leads to cohesion, leads to a general sense of everyone being on the same team. (Respondent4)

6.4.2 Improved employee satisfaction

One the most often mentioned benefit was improved employee satisfaction. Having a process where roles and responsibilities are defined more clearly, have helped people to be happier in their roles according to interviewees. This manifests in many positive ways within and outside the software product management. One respondent described it as the whole team coming together in a

way unlike anything seen earlier with high motivation. To some, the change has been even quite radical as described by Respondent8:

> Everybody want to do things right, and a lot of them have been quite frustrated with the way we have been doing things earlier […] they felt they were almost hostages having to do things they did not believe in […] they've been so happy that we've done these changes and they're so motivated now, compared to what they were earlier. (Respondent8)

In some cases, the process implementation has been done in parallel to business model changes, where the employee satisfaction improvement could be viewed as a combination of both. As an example, Respondent6 described it being much more motivational to "work on a standardized product that has a lot more impact, than a one-off project that will be a replaced five years from now". The employee satisfaction also extends beyond the core product team when cross-functional processes are improved:

> Some of our colleagues who used to think that we are sort of working against them, they now feel involved and taken... you know, we have taken them with us on the journey. (Respondent8)

One of the interviewees also mentioned that the increased employee satisfaction also shows in the company metrics:

> We measure employee engagement and we have a voice of employee every year. And last year, was a year where SPM organization has the highest rating, positive rating across all other units […] people recognize that we are working on something good and something bigger. (Respondent3)

6.4.3 Improved customer satisfaction

According to the interviewees, improvement of the software management processes increases customer satisfaction. Having more managed processes makes the customers happier, with both what they get, and when they get it:

> [The product] is more relevant for our customers, whereas before it was a bit random, you know, sometimes somebody had a good idea which turned out to be also what customers needed. But equally, we produced stuff that wasn't really any demand for. And it was taking a long time, it was very unpredictable. […] Whereas now things are much more predictable in terms of capacity and in relevance. (Respondent2)

As in the end the software product management is part of the larger business processes linking to the customers, Respondent6 went even further, and links also success of the processes directly to the customer satisfaction:

> We see a lot of happy customers for the fact that their products are so much better and richer and growing faster. And I think that the best way to define success is how

> happy your clients are and how good is your position in the market. And we improved on that a lot. (Respondent6)

6.4.4 Improved communication

Another aspect closely related to both employee and customer satisfaction, is improved communication within the company. After implementation of the software product management processes, the communication between stakeholders improves, as exemplified by Respondent2: "We understand each other much better. We talk, we share ideas, we communicate much better". Respondent1 commented that the "Improved communication [with internal stakeholders] has been one of the largest changes that we have had in the last half a year while the process improvement has been ongoing".

The improved communication also increases transparency, which contributes both internally, and further to the end customer satisfaction:

> For example, as a practical example, our sales, and technical support they can enter on behalf of customers, requirements or wishes into our database and we use that in product planning to gather any information about the stuff and then also do the planning and prioritize things. And that's transparent to the people who entered these things originally and they can better communicate to the customer (Respondent1).

Therefore improved communication also helps in managing expectations, as "People know exactly what is going to happen next and where they can find the various artefacts like the roadmap, for example" (Respondent2) and "Because the planning is sound and much better documented […] there are much clearer expectations, and everything goes smoother" (Respondent1).

According to the interviewees, one major contributing factor to the improved communication is the shared terminology introduced in the process implementation or improvement:

> It helped immensely actually to progress just by using the same language and the same understanding of concepts. […] We leverage really often the ISPMA framework in our internal trainings and workshops. (Respondent3)

6.4.5 Less defects in products

The interviewees mentioned better quality of the products as a major benefit from software management process implementation. After implementing process improvements, Respondent2 stated that "When the software out, it comes out at a higher level of quality". Respondent1 attributed the improvement to quality to the more structured planning processes:

> In terms of bugs in software, as we have now implemented something in much clearly planned than earlier, then the bug count has gone down because everybody is

> more on the same page, knows what they should do and how they should do things (Respondent1)

Respondent6 had the same view, that the improved processes increase the quality of the products they deliver:

> I think it gave us a lot of more control, and a lot of more quality in our software […] the quality of the service we deliver and the products that we deliver is much better than earlier. (Respondent6)

6.4.6 Increased customer value

In addition to making the new products and features with better quality, the interviewees also reported better outcomes, and more value delivered to their customers:

> We are not just more getting features delivered, but we're also more confident that we are building the right features. It's not just measuring outputs, but also measuring outcomes. We have enabled customers now to do certain things they couldn't do before, which align with their business needs. (Respondent2)

One of the respondents felt that earlier they did not really deliver anything of value to the customers and had high costs. The improved focus both to the correct outcomes and doing them with better quality increases the profitability, as described by Respondent8:

> We have wasted a lot of money, you know, not doing things structured, not being informed […] Now we are saving money all over the line by not having to rework things, because we're doing it right from the beginning. […] We're now actually delivering things, while earlier we weren't really delivering anything because we were constantly circling around. We didn't have a clear purpose on where we were going. (Respondent8)

6.4.7 Other benefits

In addition to the most frequently reported benefits, there were also other benefits raised by the interviewees. For example, Respondent3, whose company works in a very regulated business domain and they get audited often both internally and externally, stated that:

> I can without doubt say, that if not for SPM organization, processes and framework and standards we adopted, it would have been probably impossible to go through the audit successfully. (Respondent3)

Respondent5 and Respondent8 mentioned that in their company there is now much better understanding and management of the market requirements, and they can work in a much more structured way with sales.

Respondent8 stated that "we are effectively just working a lot faster now than we were before" and Respondent2 mentioned that in addition to the speed gains, the development schedules and content are more predictable due to the improvements. Finally, Respondent6 mentioned that their project deliveries have been 75% faster after changing their business model from professional services to standardized products and adopting the software product management processes.

6.5 Summary

Based on the quantitative and qualitative data from surveys and interviews presented in the sections above, a summary about the best practices for implementing or improving software product management processes based on the ISPMA's SPM framework, can be created. The research found several best practices, which have been categorized in high level into three success factor categories as depicted below (figure 19).

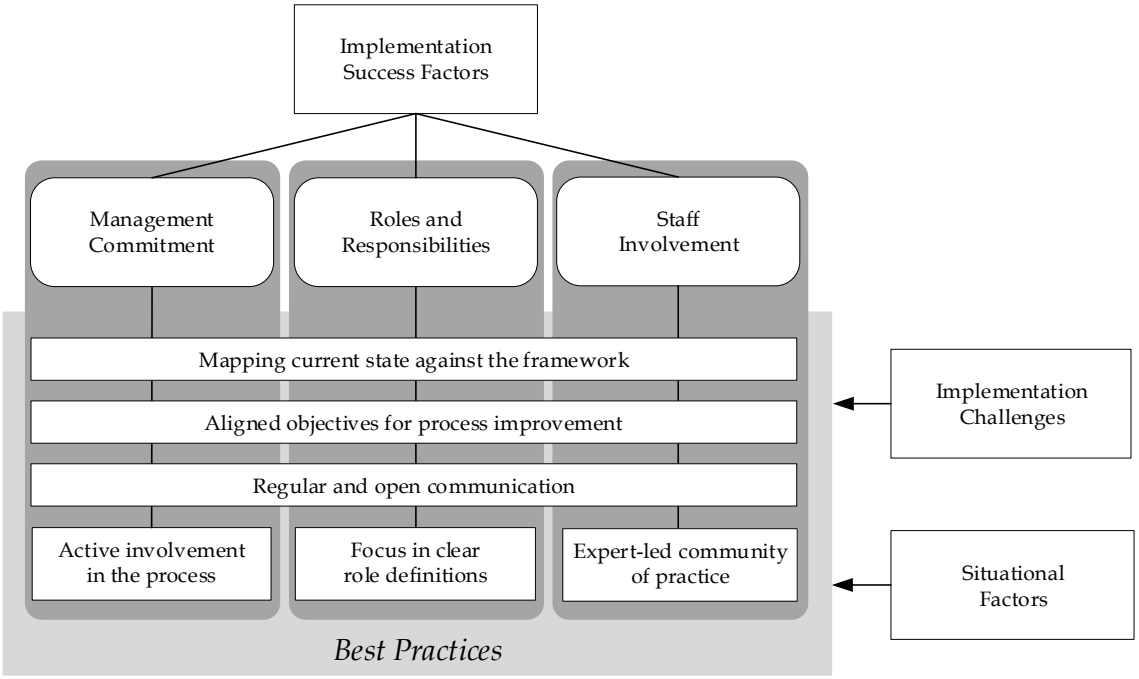

FIGURE 19 Best practices for implementing software product management processes based on the ISPMA's SPM framework

The three categories are Management Commitment, Roles and Responsibilities, and Staff Involvement, and some of the best practice reside or span over multiple categories. The best practices vary depending on organisations' situational factors, and implementation challenges associated with the process improvements. In the following, each best practice is described.

Before undertaking improvements, one needs to understand the current state to know the right order and priority of improvements. This can be accomplished by *mapping current state against the framework* through maturity assessment. To get the insight of the status quo, one must dig deep into the current ways of working by mapping the existing software product management processes, observing, discussing, and potentially also living through some challenges within the software organisation. Utilizing consultants or tools such as the Software Product Management Maturity Matrix, can be useful in establishing a comprehensive picture of the process maturity.

Both for management and staff, *aligned objectives for process improvement* are necessary for the process improvements to succeed. KPIs (Key Performance Indicators) and OKRs (Objectives and Key Results) can be useful as management tools in supporting appropriate alignment with the process improvement goals, direction, and focus.

Regular and open communication is also important for the success. From management, communication is required about the process change, progress in it, and the responsibilities related to the process change. This strengthens both management and staff commitment to the change. Within staff and to the management, there needs to be regular communication as well, to ensure that changes do take place and feedback is given, received, and considered. Witnessing, and even better - experiencing - successes when the process improvements start to take hold, has a powerful impact to all involved.

Active involvement in the process improvements is required from management for the improvement process to succeed. The involvement must be real and continuous. Ways to ensure this include process ownership and involving different managers within the actual activities of software product management, such as steering groups on the strategical level.

Process changes require often new roles or at least redefinition of existing roles. It is important have *focus in clear role definitions*, so they are understandable for both persons adopting those roles, and for everyone else involved. Use of a RACI matrix to define software product management roles can be a useful tool in understanding the roles, the potential overlap between the roles, and illustrating changes in the responsibilities. The roles need to be also resourced accordingly.

Expert-led community of practice is a powerful practice which increases staff commitment by allowing a good degree of freedom in the process improvements. By combining the collective software product management experience of subject matter experts within the company into a such group, and utilizing it for process improvements, instead of the process being "run by management", can provide good results. Such community works also as a good setting for facilitating feedback.

In addition to the best practices outlined above, some other guidance and practices were also identified in the research. Such practices include for example implementing changes in small increments, separate unit for product man-

agement to facilitate change, and widespread involvement across the organisation in the process improvements.

Different organisations have many context-dependent situational factors such as type of business, size of organisation, amount and type of software products, and level of maturity in the current processes. Such factors affect to the occurrence and application of best practices. In addition, there are several implementation challenges such as change resistance, stakeholder buy-in, sales orchestration and agile adoption, which need to be considered in the improvement process.

Regarding benefits gained by implementing or improving software product management processes based on the ISPMA's SPM Framework, the interviewees reported several different benefits. The following table (table 13) summarizes the main benefits based on the interviews:

TABLE 13 Benefits of implementing or improving software product management processes based on the ISPMA's SPM Framework

Benefit	**Type**
Increased awareness of software product management	Internal
Improved employee satisfaction	Internal
Improved customer satisfaction	External
Improved communication	Internal and External
Less defects in products	Internal and External
Increased customer value	Internal and External

The benefits are typed based on whether the impact of the benefit is more internal or external. Internal benefits have positive impact to the company internally, but do not directly show to external stakeholders such as customers. External benefits are directly noticed also by the companies' external stakeholders.

The interviewees were also asked to think of both measurable and non-measurable benefits. However, since there was so much variation how the interviewees saw the measurability, it is not shown on the summary above. In any case, based on the interviews, one could see that some of the benefits are more measurable than others, which are more subjective in nature.

In the next and final chapter of the thesis, the results are discussed from the viewpoint of the research questions, implications for both theory and practice, limitations, and possibilities for future research.

7 CONCLUSION AND DISCUSSION

This section focuses on summarizing the conducted research and discusses the implications and limitations of it. In the first section, the results of the research are summarized and discussed from the viewpoint of research problem and research questions. The second and third section discuss the implications of the research to theory and practice. In the fourth section, limitations of the research are reviewed, and the final section discusses areas for future research.

7.1 Research questions and summary

The purpose of this research was to find out what are the best practices and benefits in implementing the ISPMA's SPM Framework. The research topic was approached via the following research questions:

RQ1: What best practices have been used in different organisations in applying the ISPMA's SPM framework?
RQ2: What benefits have been achieved by applying the ISPMA's SPM framework?

The ISPMA's SPM Framework maintained by the International Software Product Management Association (ISPMA), is a framework for software product management which structures software product management practices into functional areas. Before delving deep into the research questions, a comprehensive understanding of the complex subject is required. A literature review was conducted to provide insight for what software product management is, what kind of alternatives exists for management frameworks, and what is the relevance of process improvements in software product management.

Software product management combines many different disciplines and contains multitude of activities led by a software product manager, which are to be performed over the non-predetermined lifecycle of the software product. Software products have several special characteristics unlike physical goods, which contribute into the overall challenge of software product management.

There are several different frameworks that can assist organisations in implementing software product management principles and practices. In addition of utilizing generic product management frameworks, there have been many attempts to create a framework specialized for software product management. The approaches and scope in the different frameworks differ, and based on the literature review, the ISPMA's SPM Framework takes the most holistic and balanced approach to it.

In the ISPMA's SPM Framework, maintained by the International Software Product Management Association (ISPMA), product management practices are structured into functional areas, which are further grouped into Core SPM, Participation, and Orchestration areas. Core areas include product strategy and product planning as central for software product management. Participation focuses on strategic management, while the Orchestration covers development, marketing, sales and distribution, and service and support activities. Based on the literature review, the framework is an excellent fit for software product management but provides little guidance on how to make most of it in practice.

Processes are specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs. Software Process Improvement (SPI) is a popular approach to deliver improvements in software products which contains definitions of sequence of tasks, tools, and techniques to be performed to plan and implement improvement activities. Process improvements in software product management, however, have been a relatively low focus area until recently. Majority of the success factors and barriers for process improvement revolve naturally around human factors, such as commitment, skills and process improvement understanding, since the software processes are human-centered activities. There are both generic approaches, but also empirically proven methods, such as Software Product Management Competence Model and Software Product Management Maturity Matrix, to improve the software product management processes significantly. The context of the software organisation impacts which processes should be improved and in which order, and such situational factors must be considered in the improvement process.

To address research questions RQ1 and RQ2, a mixed-method research approach was applied to find out the best practices and benefits associated to implementing processes based on the ISPMA's SPM Framework. The research was done using quantitative survey and semi-structured qualitative interview research methods to framework practitioners accessed via the ISPMA organisation. The target of the survey was to produce demographics of the worldwide ISPMA's SPM Framework practitioners, and views on what kind of practices are or could be used when implementing or improving processed based on the framework. The purpose of the conducted interviews was to gain in-depth insight about the best practices and benefits from selected practitioners, based on the interview structure influenced by the background literature and the survey results.

Research question RQ1 revolved around best practices which contribute to the success of implementing or improving processes based on the ISPMA's SPM Framework. The research found several best practices, which can be roughly grouped into three categories: management commitment, staff involvement, and roles and responsibilities. Best practices overarching several categories include mapping current state against the framework, aligned objectives for process improvement, and regular and open communication. For understanding the current state, in addition to dialogue and observation, tools such as Software Product Management Maturity Matrix can be utilized. Similarly, use of KPIs and OKRs can be helpful in the objective alignment. In the management commitment category, it is crucial that the management has active involvement in the process improvements. For roles and responsibilities, it is important that there is focus in clear definitions of the roles related to the process. Tools such as RACI matrix can be useful in establishing common understanding of the roles and changes. Finally, for staff involvement category, running an expert-led community of practice is a powerful practice to encourage feedback, utilize best possible experience, and increase staff involvement to the changes. In addition to the main best practices listed above, other less frequent practices and guidelines were also identified in the research. The best practices can be situation dependent, and such situational factors need to be considered when applying the practices. Success of the process improvement can be also hindered by various implementation challenges, which practitioners should be aware of.

Research question RQ2 was about benefits for implementing the ISPMA's SPM Framework. The research found that there are several benefits for implementing or improving processes based on the framework, which have positive impact to the company internally, but also to external stakeholders such as customers. The internal benefits include increased awareness of software product management and improved employee satisfaction. Benefits having both internal and external impact include improved communication, less defects in products, and increased customer value. In addition, improved customer satisfaction was identified as an external benefit. In addition to the main benefits listed above, also other minor benefits were found, and some of the interviewees reported that the benefits had a directly measurable aspect.

7.2 Implications to theory

The findings in the research contribute to the overall knowledge about what kind of best practices can be utilized in organisations when implementing or improving software product management practices, both in general and specifically in the context of the ISPMA's SPM Framework.

One of the found best practices in this research is understanding the current state of the practices. This matches previous research by Ebert and Brinkkemper (2014) where the use of assessments is highlighted as important during the first stages of the change process. Similarly, research by Galinac

(2009) states that assessments are important input for creation of improvement backlog and project specification in the first two phases of process improvement.

Additionally, based on this research, there seems to be a demand for tools for assessment and selection of next steps in process improvement. One of the interviewees had been involved with the Software Product Management Maturity Matrix development research (Bekkers et al., 2012), and reported being satisfied with it, which supports the applicability of the matrix also to the ISPMA's SPM Framework.

Aligned objectives within organisation for process change have not been much of discussion concerning process improvement in software product management or in software process improvements. However, research by Ebert and Brinkkemper (2014), acknowledges KPIs as a component for start of the change process. This is also aligned with Galinac's (2009) inclusion of monitoring and measurement as best practices for process improvement.

Regular and open communication was one of the found best practices. This matches well with Kotter's (1996) fourth stage of leading change and is in line with the study by Ebert and Brinkkemper (2014), who highlights communication as important for the roll-out step of the change process. This also relates closely to a minor finding in the study that process improvements should be done in small increments. Weerd et al. (2010) agree that gradual process improvements are important for most organisations but note that small increments require much communication. However, in Galinac's (2009) view, implementing changes in small increments and deliverables improves participation, but simplifies change management in the process improvements.

The identified best practices of active involvement in the process improvements, and utilization of expert-led communities of practice, are aligned with findings by Weerd et al. (2010b) in the context of global software product management, and more generally with Kotter's (1996) view on empowering change. The findings are also aligned with Ebert and Brinkkemper (2014), who have reported that management commitment is important throughout the change process, and that the feedback loop from product management community is important for identifying further improvement needs.

Defining clear roles concerning the software product management roles and responsibilities, was a major best practice found in this study, which can be assisted by practical tools such as RACI matrices. This need for clear role setting is also confirmed by Ebert and Brinkkemper (2014), who see the role definition as a clear activity within the first phases of the change process.

Based on the heterogeneity of the respondents and their organisations, an observation was also made that the best practices are context dependent and affected by situational factors. This aligns well with earlier research by Bekkers et al. (2008) and Bekkers (2012), where different situational factors have been identified to not only affect daily practices, but also capability implementation and method selection for software product management.

Based on the quantitative survey, the three most important success factors for process improvement are management commitment, roles and responsibili-

ties, and staff involvement. The findings align well with management commitment, which was also the most significant success factor for SPI (Software Process Improvement) according to Khan et al. (2017). It is also worth noting that none of the success factors was ranked as "Not important" by any of the respondents, and the differences were not large, especially considering the sample size. An opposite of success factors is barrier, or a challenge. The challenges found in this study are partially comparable to barriers reported by Khan et al. (2017) and Khan at al. (2018) for SPI. With that in mind, it could be cautiously said that the success factors for SPI are most likely applicable also to process improvements concerning software product management, but further confirmatory studies are required.

Concerning the benefits for implementing or improving software product management practices, the results of this study are primarily in line with what has been found in previous research. Found benefits of improving processes, such as less defects in products, improved customer satisfaction, and increased customer value are fully aligned with findings by Ebert (2007), Ebert and Brinkkemper (2014), Hall et al. (2002), and Goldenson & Herbsleb (1995). Some of these benefits are also measurable, as reported in previous research by Ebert (2007) and Ebert and Brinkkemper (2014) in their longitudinal research. The measurability of some the benefits also manifested itself in this study, although it was not further pursued. However, in previous research, internal benefits such as improved employee satisfaction, improved communication, and overall increased awareness of software product management, have not been in focus. It could be due to their nature or the perceived non-measurability, but these findings could be attributed as new knowledge provided by this research.

As a summary, the contribution of this research to theory concerning best practices is mostly a new consolidated view of the best practices and a confirmation of existing practices, in the new context of the ISPMA's SPM Framework. For benefits, the contribution of the research is a list of benefits, which of a few seem to be previously unexplored by past studies.

7.3 Implications to practice

The findings in this thesis have several implications for practice, which could be utilized by different organisations or individuals. The best practices found in the research could be utilized by companies by ensuring that those are considered in planning, implementing, and improving processes based on the ISPMA's SPM Framework. Especially already available concrete tools, such as the Software Product Management Maturity Matrix and different RACI Matrices, could be of great help. Utilizing the maturity matrix can speed up identifying the gaps and the right areas to focus to, while RACI matrices can assist in role definition and communicating the changes.

ISPMA organisation could use the found best practices for basis of creating practical guides for implementation and improvement activities based on

the ISPMA's SPM Framework, to facilitate successful framework adoption. Such guides could and should also consider the identified challenges and provide guidance how to navigate around them. The same could also be done by the various consultation and training companies focusing on the framework, instead of ISPMA, but the effect of such work would be most likely more limited in impact.

The demographics collected about framework practitioners via the survey can also provide value to the ISPMA organisation. Although the results are not completely representative of all members, the descriptive statistics could be useful in planning further development of the framework, or for example, event organisation.

The benefits found regarding implementation or improvement of software product management processes could be also utilized in practice. For example, the benefits could be of use in organisations when building metrics for process implementation or improvement. Concrete metrics could be developed from the benefits for following the success and proceeding of the process improvement efforts within an organisation. In similar fashion, the benefits could be utilized in the training and consultation related to the ISPMA's SPM Framework and its application.

Since several benefits were found, they also support the more general applicability of the framework for software product management. Although the selection bias of the research must be considered, the views of the respondents towards the framework were primarily favourable and in support to the view, that the ISPMA's SPM Framework is among the most suitable frameworks for improving software product management practices. The benefits themselves and the applicability of the framework could be of limited use for example in marketing efforts for software product management training either by ISPMA, or training and consultation providers.

7.4 Limitations

As every research, this too had some limitations of its own. In the literature review, it is possible that relevant studies were missed, due to time constraints for the research and the fact that software product management is multidisciplinary. This could be of influence on the data collection and analysis of the performed quantitative and qualitative research.

The purpose of quantitative research is to allow generalization from a research sample to the whole population, which the sample represents. This study used a small group of participants compared to the population, which in combination with the potential selection bias in the voluntary response sampling, limits the generalizability of the survey results to the whole ISPMA community of software product management practitioners. This is also the case in global approach – while in the survey there were responses received from individuals from 11 countries, in many cases there was only one respondent

from a certain country, and many countries are absent to represent a truly global view. Therefore, the findings in the quantitative part of the research cannot be generalized.

When conducting an online survey, it is difficult for the researcher to know, how the respondent took the survey, if they have answered truthfully, and if the questions have been correctly interpreted. In attempt to control the collection of research data the link to the survey was sent only to the ISPMA certified members email list, but with an online survey, one can never be sure who answered the survey. This was not such an issue for the qualitative interviews, as the semi-structured interview method allowed dialogue to ensure correct interpretation of questions and answers. The interviewees experienced relatively similar interview settings due to remote interviewing method. Although their own interviewing environment during the sessions varied, all interviewees were familiar and routinely using such tools in their daily work.

The survey respondents and interviewees were primarily from large or mid-sized companies. The lack of small companies and start-ups may indicate that the study does not represent the full spectrum of software companies, that may utilize the ISPMA's SPM Framework today. In any case, it does not represent the whole software industry and the potential users of the framework. It should be also noted that the survey and interviews were targeted only to the ISPMA certified product management professionals, which answered only from their point of view regarding the best practices in their respective organisations. By interviewing also other people within those organisations would have given further strength to the findings.

In summary, the combined results of the mixed methods utilized in this study are not generalizable. However, the best practices and benefits found in the study can be transferable and utilized in other contexts, providing that the limitations described above are taken into consideration. In addition, the developed interview themes and predefined questions could be utilized for further studies.

7.5 Future research

In future research, there is much room for both quantitative and qualitative studies. It is valuable to study further, what are the best practices and benefits for implementing the ISPMA's SPM Framework and other frameworks, to increase understanding of what makes implementation of such value-increasing practices smoother and more effective.

Implementing and especially improving the software management processes within an organisation takes a long time. A longitudinal study on the best practices of improving software product management processes and reaping benefits over a long period could provide further information on whether the practices change over time, and which improvement steps provide the most benefits.

Future research focused on small companies and start-ups could be of interest, to find out how software product management frameworks, or specifically the ISPMA's SPM Framework, fit into such contextual settings. The study could find new best practices or define new needs for the frameworks and processes themselves that are more applicable to smaller settings.

Similarly, the relationship between "traditional" software product management and agile practices, could be worthwhile to study. Information on how the software management practices work in different agile settings could provide important information required for compatibility and development of different approaches.

The uncovered internal benefits of implementing software product management processes could also be an interesting area for future research. In addition to mapping out more "hard" and "measurable" benefits related to the company and product success, it might be interesting to investigate the more subjective side of benefits – for example employee satisfaction and improved communication – in effort to find out how those benefits manifest, contribute to the overall success, and could be measured.

Regardless of the setting, the need for evaluating the current state of software product management practices is necessary before improvement efforts. Future research, which would create or update a software process management maturity assessment tool specifically for the ISPMA's SPM Framework, matching all activities within, could be of value and have both practical and scientifical applications. In addition, research could also be made about how the perceived maturity level of software product management correlates with improvement areas.

Since this research focused only on one framework, and all frameworks are different, interesting results could be found by researching how implementation practices for more formalized process frameworks differ from more relaxed approaches. Such research might provide information on how those differences reflected to the best practices and benefits.

REFERENCES

Abran, A. & Moore, J. M. (2004). *Software Engineering Body of Knowledge*. California, IEEE Computer Society.

Adams, J., Khan, H., & Raeside, R. (2014). *Research methods for business and social science students*. New Delhi: SAGE Publications India.

Agile Manifesto (2001). *Manifesto for Agile Software Development*. Accessed 29.12.2019 http://agilemanifesto.org

AIPMM (2017). *The Seven-Phase Product Life Cycle*. Accessed 3.2.2020 https://www.slideshare.net/aipmm/aipmm-phase-review-process-for-product-management

AIPMM (2020a). *What we do*. Accessed 3.2.2020 https://aipmm.com/what-we-do

AIPMM (2020b). *The Certified Product Manager Credential*. Accessed 3.2.2020 https://aipmm.com/cpm

AIPMM (2020c). *The Guide to the Product Management & Marketing Body of Knowledge*. Accessed 3.2.2020 http://productmanagementbook.com/

Anderson, C. & Zeithaml, P. (1984). Stage of the Product Life Cycle, Business Strategy, and Business Performance. *Academy of Management Journal*, 27(1), 5-24.

Artz, P., van de Weerd, I,. Brinkkemper, S. & Fieggen, J. (2010). Productization: Transforming from Developing Customer-Specific Software to Product Software. *Proceedings of the 1st International Conference on Software Business* (ICSOB 2010), 90-102.

Aysolmaz, B. & Demirörs, O. (2011). A Detailed Software Process Improvement Methodology: BG-SPI. *European Conference on Software Process Improvement*, 97-108. Springer, Berlin, Heidelberg.

Bekkers, W. (2012). *Situational Process Improvement in Software Product Management* (Dissertation). Utrecht University.

Bekkers, W., van de Weerd, I., Brinkkemper, S. & Mahieu, A. (2008). The Influence of Situational Factors in Software Product Management: An Empirical Study. *Second International Workshop on Software Product Management (IWSPM 2008)* (41-48). IEEE.

Bekkers, W., van de Weerd, I., Spruit, M. & Brinkkemper, S. (2010). A Framework for Process Improvement in Software Product Management. *European systems & software process improvement and innovation (EuroSPI 2010)*, 1-12. Berlin: Springer.

Bekkers, W., Brinkkemper, S., van den Bemd, L., Mijnhardt, F., Wagner, C. & van de Weerd, I. (2012). Evaluating the Software Product Management Maturity Matrix. *2012 20th IEEE International Requirements Engineering Conference (RE)*, 51-60. IEEE.

Benaroch, M. & Appari, A. (2010). Financial Pricing of Software Development Risk Factors. *IEEE Software*, 27(5), 65-73.

Blackblot (2019a). Blackblot Product Manager's Toolkit (PMTK) Methodology. Accessed 26.12.2019 https://www.blackblot.com/methodology

Blackblot (2019b). Blackblot Product Manager's Toolkit (PMTK) Overview. Accessed 26.12.2019 https://www.blackblot.com/files/pmtk/Blackblot_PMTK_Evaluation/PMTK/Blackblot_PMTK_Overview.pdf

Blackblot (2019c). Blackblot - International Premium Brand. Accessed 26.12.2019 https://www.blackblot.com/international

Blackblot (2019d). Blackblot Strategic Product Manager™ Course. Accessed 26.12.2019 https://www.blackblot.com/spmr

Blackblot (2019e). Blackblot Strategic Product Management™ Training Program Curriculum. Accessed 26.12.2019 https://www.blackblot.com/education

Blackblot (2019f). Blackblot PMTK Methodology™ Implementation. Accessed 26.12.2019 https://www.blackblot.com/implementationf

Blackblot (2019g). PMTK Action Model. Accessed 26.12.2019 https://www.blackblot.com/files/pmtk/Blackblot_PMTK_Evaluation/PMTK/PMTK_Methodology/PMTK_Core_Models/Blackblot_PMTK_Action_Model.pdf

Cabanillas, C., Resinas, M., & Ruiz-Cortés, A. (2011). Mixing RASCI matrices and BPMN together for responsibility management. *VII Jornadas en Ciencia e Ingeniería de Servicios (JCIS'11), 1167*.

Camp, R. (1989). *Benchmarking: The search for industry best practices that lead to superior performance*. Milwaukee: ASQ Quality Press.

Chisa, E. (2014). Evolution of the Product Manager. *Communications of the ACM,* 57(11), 48-52.

Collabnet. (2019). 13th annual State of Agile Report. Accessed 19.12.2019 https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508

Colomo-Palacios, R., Soto-Acosta, P, García-Peñalvo, F. J. & Carcía-Crespo, Á. (2012). A Study of the Impact of Global Software Development in Packaged Software Release Planning. *Journal of Universal Computer Science*, 18(19), 2646-2668.

Conway, M. (1968). How Do Committees Invent? *Datamation*, 14(4), 23-81.

Davenport, T. H. (1993). *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press, Boston MA.

Dekkers, C. & Forselius, P. (2007). Increase ICT Project Success with Concrete Scope Management. *Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications*, 385-392, Los Alamitos: IEEE.

Ebert, C. (2007). The Impacts of Software Product Management. *The Journal of Systems and Software*, 80(6), 850-861.

Ebert, C. (2009). Software Product Management. *CrossTalk, The Journal of Defence Software Engineering*, 22, 15-19.

Ebert, C. (2014). Software Product Management. *IEEE Software,* 31(3), 21-24.

Ebert, C. (2018). Managing software products in a global context. *Proceedings of the 13th International Conference on Global Software Engineering,* 69-76. Stuttgart: Vector Consulting Services.

Ebert, C. & Brinkkemper, S. (2014). Software product management – An industry evaluation. *The Journal of Systems and Software*, 95, 10-18.

Ebert, C. & Paasivaara, M. (2017). Scaling agile. *IEEE Software*, 34(6), 98-103.

Etikan, I., Musa, S. A., & Alkassim, R. S. (2016). Comparison of convenience sampling and purposive sampling. *American journal of theoretical and applied statistics*, 5(1), 1-4.

EuroAsia SPI (2019). *SPI Manifesto*. Accessed 19.1.2020 http://2018.eurospi.net/index.php/manifesto

Fox, H. (1973). A Framework for Functional Coordination. *Atlanta Economic Review*, 23(6), 8-11.

Fricker, S. (2012). Software product management. In Maedche, A., Botzenhardt, A. & Neer L. (Eds.), *Software for People,* (53–81). Berlin: Springer-Verlag.

Galinac, T. (2009). Empirical evaluation of selected best practices in implementation of software process improvement. *Information and Software Technology*, 51(9), 1351-1364.

Gartner (2018). *Digital business requires a move from project to product-centric business models*. Accessed 29.12.2019 https://www.gartner.com/smarterwithgartner/cio-agenda-2019-move-from-project-to-product-delivery/

Gartner (2019a). *Gartner Says Global IT Spending to Grow 0.6% in 2019*. Accessed 28.12.2019 https://www.gartner.com/en/newsroom/press-releases/2019-10-07-gartner-says-global-it-spending-to-grow-06-in-2019

Gartner (2019b). *Gartner Survey Finds 85 Percent of Organizations Favor a Product-Centric Application Delivery Model*. Accessed 29.12.2019 https://www.gartner.com/en/newsroom/press-releases/2019-02-19-gartner-survey-finds-85-percent-of-organizations-favor-a-product

Geracie, G., Eppinger, S. (2013). *The Guide to the Product Management and Marketing Body of Knowledge: ProdBOK Guide.* Product Management Educational Institute.

Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorscheck, T. & Abrahamsson, P. (2016). Software Development in Startup Companies: The Greenfield Startup Model. *IEEE Transactions on Software Engineering*, 42(6), 585-604

Goldenson, D. & Herbsleb, J. (1995). *After the Appraisal: A Systematic Survey of Process Improvement, Its Benefits, And Factors That Influence Success*. SEI: CMU/SEI-95-TR-009. Software Engineering Institute, USA.

Gorchels, L. (2011). *The Product Manager's Handbook, Fourth Edition*. New York, McGraw-Hill.

Gorschek, T., Gomes, A., Pettersson, A. & Torkar, R. (2012). Introduction of a process maturity model for market-driven product management and requirements engineering. *Journal Of Software Maintenance And Evolution*, 24(1), 83-113.

Haines, S. (2009). *The Product Manager's Desk Reference, Second Edition*. New York, McGraw-Hill.

Hall, T., Rainer, A. & Baddoo, N. (2002). Implementing software process improvement: An empirical study. *Software Process: Improvement and Practice*, 7(1), 3–15.

Harkonen, J., Haapasalo, H. & Hanninen, K. (2015). Productisation: A review and research agenda. *International Journal of Production Economics*, 164, 65-82.

Herbsleb, J. & Grinter, R. (1999). Architectures, coordination, and distance: Conway's law and beyond. *IEEE Software*, 16(5), 63-70.

Hirsjärvi, S., Remes, P. & Sajavaara, P. (2009) *Tutki ja Kirjoita*. Helsinki: Tammi.

Horowitz, B. (1998). *Good Product Manager/Bad Product Manager*. Accessed 4.1.2019 https://a16z.com/2012/06/15/good-product-managerbad-product-manager/

Humphrey W. S. (1989). *Managing the Software Process*. Boston, Massachusetts: Addison-Wesley Longman Publishing Co.

ISO. (2019). *ISO/IEC 33001:2015(en) Information technology — Process assessment — Concepts and terminology*. Accessed 20.1.2020 https://www.iso.org/obp/ui/#iso:std:iso-iec:33001:ed-1:v1:en

ISPMA. (2016a). *SPM Syllabus - Foundation Level*. Accessed 8.2.2020. https://ispma.org/wp-content/uploads/2014/12/ISPMA-SPM-FL-Syllabus-V.1.3.pdf

ISPMA. (2016b). *SPM Syllabus - Excellence Level – Product Strategy*. Accessed 8.2.2020. https://ispma.org/wp-content/uploads/2014/12/ISPMA-SPM-EL-Syllabus-ProductStrategy-V.1.1-Student-Edition.pdf

ISPMA. (2016c). *SPM Syllabus - Excellence Level – Product Planning*. Accessed 8.2.2020. https://ispma.org/wp-content/uploads/2014/12/ISPMA-SPM-EL-Syllabus-Product-Planning-V1.1-Student-Edition.pdf

ISPMA. (2016d). *SPM Syllabus - Excellence Level – Strategic Management*. Accessed 8.2.2020. https://ispma.org/wp-content/uploads/2014/12/ISPMA-SPM-EL-Syllabus-StrategicManagement-V.1.2-Student-Edition.pdf

ISPMA. (2017). *SPM Syllabus - Excellence Level – Orchestration*. Accessed 8.2.2020. https://ispma.org/wp-content/uploads/2014/12/ISPMA-SPM-EL-Syllabus-Orchestration-V.1.0-Student-Edition.pdf

ISPMA. (2019a). *Certification*. Accessed 4.1.2020 https://ispma.org/training-and-certification/certification-partners/

ISPMA. (2019b). *Syllabi*. Accessed 4.1.2020 https://ispma.org/body-of-knowledge/syllabi/

ISPMA. (2019c). *Certified Members*. Accessed 4.1.2020 https://ispma.org/membership/certified-members/

Ivankova, N., Creswell, J., & Stick, S. (2006). Using mixed-methods sequential explanatory design: From theory to practice. *Field methods*, 18(1), 3-20.

Khan, A., Keung, J., Hussain, S, Niazi, M. & Tamimy, M. (2017). Understanding software process improvement in global software development: a theoretical framework of human factors. *ACM SIGAPP Applied Computing Review*, 17(2), 5-15.

Khan, A., Keung, J., Niazi, M., Hussain, S. & Shameem, M. (2018). GSEPIM: A roadmap for software process assessment and improvement in the domain of global software development. *Journal of Software: Evolution and Process*, 31(1).

Kitchenham, B., & Pfleeger, S. L. (2003). Principles of survey research part 6: data analysis. *ACM SIGSOFT Software Engineering Notes*, 28(2), 24-27.

Kittlaus, H.-B. (2012). Software Product Management and Agile Software Development: Conflicts and Solutions. In Maedche, A., Botzenhardt, A. & Neer L. (Eds.), *Software for People,* (83–96). Berlin: Springer-Verlag.

Kittlaus, H.-B. & Clough, P. (2009). *Software Product Management and Pricing: Key Success Factors for Software Organizations*. Springer.

Kittlaus, H.-B. & Fricker, S. (2017). *Software Product Management: The ISPMA-Compliant Study Guide and Handbook*. Springer.

Kotler, P., Armstrong, G. (1994). *Principles of Marketing (6th Edition)*. Englewood Cliffs, N.J. Prentice Hall.

Kotler, P., Armstrong, G., Harris, L., Piercy, N. (2017). *Principles of Marketing (7th European Edition)*. Pearson Education Limited. Harlow.

Kotter, J. (1996). *Leading change*. Boston: Harvard Business School Press.

Kuhrmann, M., Diebold, P., & Münch, J. (2016). Software process improvement: a systematic mapping study on the state of the art. *PeerJ Computer Science*, *2*, e62.

Leffingwell, D. (2011). *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley Professional.

LUT University. (2020). *Master's Programme in Software Product Management and Business*. Accessed 3.2.2020 https://www.lut.fi/web/en/admissions/masters-studies/msc-in-

technology/software-engineering/software-product-management-and-business

Maglyas, A. (2013). *Overcoming the complexity of software product management* (Dissertation). Lappeenranta University of Technology.

Maglyas, A., Nikula, U. & Smolander, K. (2013). What are the roles of software product managers? An empirical investigation. *Journal of Systems and Software,* 86(12), 3071–3090.

Maglyas, A., Nikula, U., Smolander K. & Fricker S. (2017). Core software product management activities. *Journal of Advances In Management Research*, 14(1), 23-45.

Manteli, C., van de Weerd, I. & Brinkkemper, S. (2010). Bridging the gap between software product management and software project management. *Proceedings of the 11th International Conference on Product Focused Software*, 32-34.

Mkrtchyan, R. (2019). *The History and Evolution of Product Management (Part 3).* Accessed 12.1.2020 https://medium.com/pminsider/the-history-and-evolution-of-product-management-part-3-c91698c889ca

Moore, G. (1991). *Crossing the chasm: Marketing and selling technology products to mainstream customers*. New York: HarperBusiness.

Moore, G. (1994). Crossing the customer chasm. *Upside*, Vol 6, Foster City.

Myers, M. & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and organization*, 17(1), 2-26.

Niazi, M., Wilson, D. & Zowghi, D. (2006). Critical success factors for software process improvement implementation: an empirical study. *Software Process Improvement and Practice*, 11(2), 193-211.

Niazi, M., Babar, M. & Verner, J. (2010). Software Process Improvement barriers: A cross-cultural comparison. *Information and Software Technology*, 52(11), 1204-1216.

Pancholi, A. & Grover, S. (2014). Scaled agile framework: a blight. *International Journal of Innovative Research and Development (IJIRD)*, 3(5), 425-427.

Patton, M. Q. (1990). *Qualitative research & evaluation methods, Second Edition*. Sage publications, Inc.

Pinsonneault, A., & Kraemer, K. (1993). Survey research methodology in management information systems: an assessment. *Journal of management information systems*, 10(2), 75-105.

Pollack, J. & Pollack, R (2015). Using Kotter's Eight Stage Process to Manage an Organisational Change Program: Presentation and Practice. *Systemic Practice and Action Research*, 28(1), 51-66.

Popp, K. & Meyer, R. (2010). *Profit from Software Ecosystems: Business Models, Ecosystems and Partnerships in the Software Industry*. Norderstedt: Books on Demand.

Pragmatic Institute. (2019a). *Pragmatic Framework*. Accessed 2.1.2020 https://cdn.agilitycms.com/pragmatic-marketing-v2/FrameWork_2019.pdf

Pragmatic Institute. (2019b). *Pragmatic Framework*. Accessed 2.1.2020 https://www.pragmaticinstitute.com/framework

Pragmatic Institute. (2019c). *Pragmatic Framework – Framework Roles & Responsibilities*. Accessed 2.1.2020 https://cdn.agilitycms.com/pragmatic-marketing-v2/Framework_Roles.pdf

Pragmatic Institute. (2019d). *Pragmatic Institute – About Us*. Accessed 2.1.2020 https://www.pragmaticinstitute.com/about-pragmatic

Pragmatic Institute. (2019e). *Complete Training For Every Aspect of Your Business*. Accessed 2.1.2020 https://www.pragmaticinstitute.com/Courses

Pragmatic Institute. (2019f). *Frequently Asked Questions*. Accessed 2.1.2020 https://www.pragmaticinstitute.com/faq

Pries-Heje, J. & Johansen, J. (2010). *SPI Manifesto*. Accessed 19.1.2020 http://2018.eurospi.net/index.php/manifesto

Product Focus. (2019). *Product Management and Product Marketing survey*. Accessed 28.12.2019 https://www.productfocus.com/wp-content/uploads/2019/03/Product-Focus-Industry-Survey-2019.pdf

Product Management Buzz. (2019). *Product Life Cycle*. Accessed 26.12.2019 https://productmanagement.buzz/index.php/2018/02/19/better-product-outcomes-product-life-cycle/

Product Management Festival. (2019). *Product Management: Trends and Benchmarks Report 2019*. Accessed 9.12.2019 https://productmanagementfestival.com/survey/

Ramesh, L. & Clear, T. (2018). Enhancing product and service capability through scaling agility in a global software vendor environment. *Proceedings of the 13th Conference on Global Software*, 59-68. ACM.

Reeves, B & Gaines, B. (2018). *Building Products for the Enterprise: Product Management in Enterprise Software*. California: O'Reilly Media.

Scaled Agile. (2020a). *Welcome to Scaled Agile Framework® 5.0!* Accessed 3.2.2020 https://www.scaledagileframework.com/about/

Scaled Agile. (2020b). *White Paper: Achieving Business Agility with SAFe® 5.0*. Accessed 3.2.2020 https://www.scaledagile.com/resources/safe-whitepaper/

Scaled Agile. (2020c). *SAFe® for Lean Enterprises 5.0*. Accessed 3.2.2020 https://www.scaledagileframework.com/safe-for-lean-enterprises/

Scaled Agile. (2020d). *SAFe Implementation Roadmap*. Accessed 3.2.2020 https://www.scaledagileframework.com/implementation-roadmap/

Scaled Agile. (2020e). *Join the Community*. Accessed 3.2.2020 https://www.scaledagile.com/join-the-community/

Scaled Agile. (2020f). *SAFe® Course Descriptions*. Accessed 3.2.2020 https://www.scaledagile.com/certifications/which-course-is-right-for-me/

Schief, M. & Buxmann, P. (2012). Business Models in the Software Industry. *2012 45th Hawaii International Conference on System Sciences*. IEEE.

Schwaber, K & Sutherland, J. (2017). *The Scrum Guide*. Accessed 29.12.2019 https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf

Smith, M. L., Erwin, J., & Diaferio, S. (2005). Role & responsibility charting (RACI). In *Project Management Forum (PMForum),* 5.

Šmite, D., Moe, N. & Ågerfalk, P. (2010), Fundamentals of Agile Distributed Software Development. In Šmite, D., Moe, N. & Ågerfalk, P. (Eds.), *Agility Across Time and Space*, (3-7). Springer.

SPM Body of Knowledge. (2019). Accessed 10.11.2019 https://ispma.org/body-of-knowledge/

SPM Body of Knowledge. (2020). Accessed 3.2.2020 https://ispma.org/body-of-knowledge/

Springer, O. & Miler, J. (2018). The Role of a Software Product Manager in Various Business Environments. *Proceedings of the Federated Conference on Computer Science and Information Systems*, 15, 985–994.

Steinhardt, G. (2017). *The Product Manager's Toolkit: Methodologies, Processes, and Tasks in Technology Product Management*. Springer International Publishing.

Taylor, F. W. (1919). *The Principles of Scientific Management*. New York: Harper & Brothers Publishers.

Teddlie, C., & Yu, F. (2007). Mixed methods sampling: A typology with examples. *Journal of Mixed Methods Research*, 1(1), 77-100.

Trkman, P. (2010). The critical success factors of business process management. *International journal of information management*, 30(2), 125-134.

Turetken, O., Stojanov, I., & Trienekens, J. (2017). Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework. *Journal of Software: Evolution and process*, 29(6).

Tyrväinen, P., Warsta, J. & Seppänen, V. (2008). Evolution of Secondary Software Businesses: Understanding Industry Dynamics. In León, G., Bernardos, A., Casar, J., Kautz, K. & DeGross, J. (Eds.), *IFIP International Federation for Information Processing, Open IT-based innovation: Moving Towards Cooperative IT Transfer and Knowledge Diffusion*, 287, 281–401. Boston: Springer.

Unterkalmsteiner, M., Gorschek, T., Islam, A. M., Cheng, C., Permadi, R. & Feldt, R. (2012). Evaluation and Measurement of Software Process Improvement — A Systematic Literature Review. *IEEE Transactions on Software Engineering*, 38(2), 398–424.

Valtakoski, A. & Rönkkö, M. (2010). Diversity of Business Models in Software Industry. *Proceedings of the 1st International Conference on Software Business* (ICSOB 2010), 1-12.

van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J. & Bijlsma, L. (2006). On the Creation of a Reference Framework for Software Product Management: Validation and Tool Support. *International Workshop on Software Product Management (IWSPM'06)*, Utrecht University, Department of Information and Computing Sciences.

van de Weerd, I. & Katchow, R. (2009). On the integration of software product management with software defect management in distributed environments. *2009 5th Central and Eastern European Software Engineering Conference in Russia (CEE-SECR)*, IEEE, 167-172.

van de Weerd, I., Bekkers, W. & Brinkkemper, S. (2010a). Developing a Maturity Matrix for Software Product Management. *Proceedings of the 1st International Conference on Software Business* (ICSOB 2010), 76-89.

van de Weerd, I., Brinkkemper, S., Versendaal, J. (2010b). Incremental method evolution in global software product management: A retrospective case study. *Information and Software Technology*, 52(7), 720-732.

Venkatesh, V., Brown, S. A., & Bala, H. (2013). Bridging the Qualitative-Quantitative divide: Guidelines for conducting Mixed Methods research in information systems. *MIS Quarterly*, 37(1), 21-54.

Vizcaíno, A., García, F., Piattini, M. & Beecham, S. (2016). A validated ontology for global software development. *Computer Standards & Interfaces*, 46, 66-78.

Vähäniitty, J. (2012). *Towards agile product and portfolio management*. (Dissertation). Aalto University.

Willman, P. (1996). *Organizational Change in the 1990's*, Lecture, University of Limerick, Ireland.

Xu, L. & Brinkkemper, S. (2007). Concepts of product software. *European Journal of Information Systems*, 16(5), 531-541.

APPENDIX 1: PRODUCT MANAGEMENT FRAMEWORKS

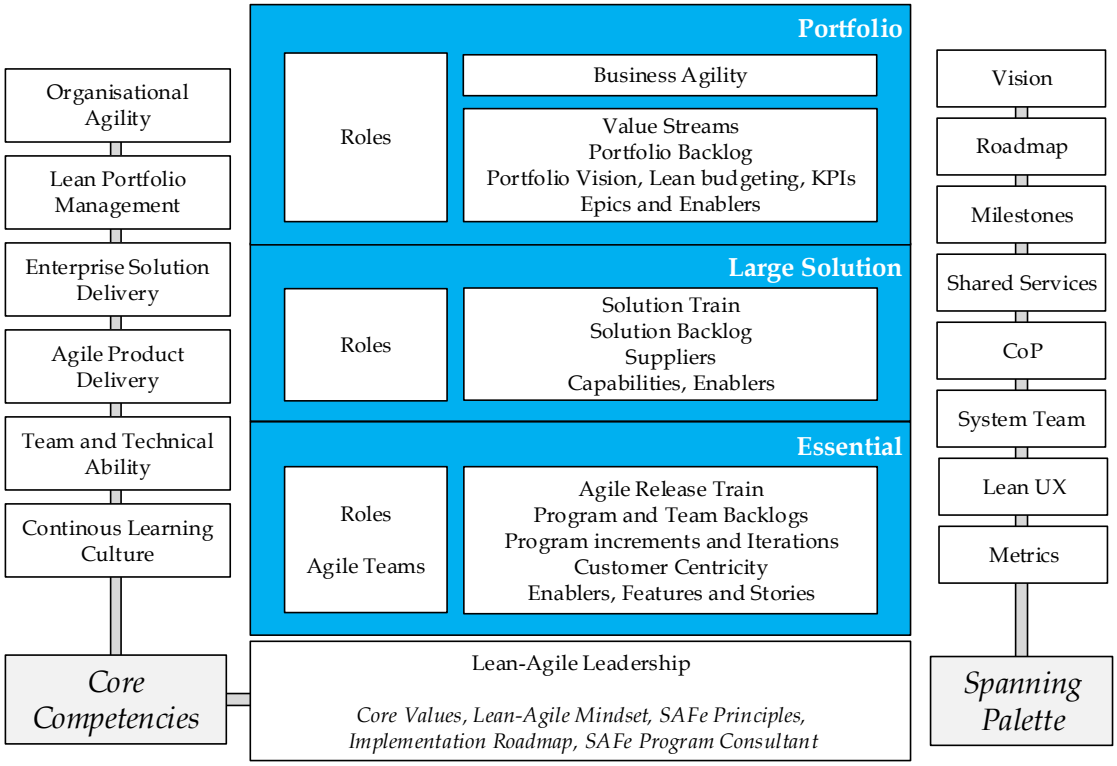

FIGURE: Full Scaled Agile Framework "Big Picture" (Adapted from Scaled Agile, 2020c)

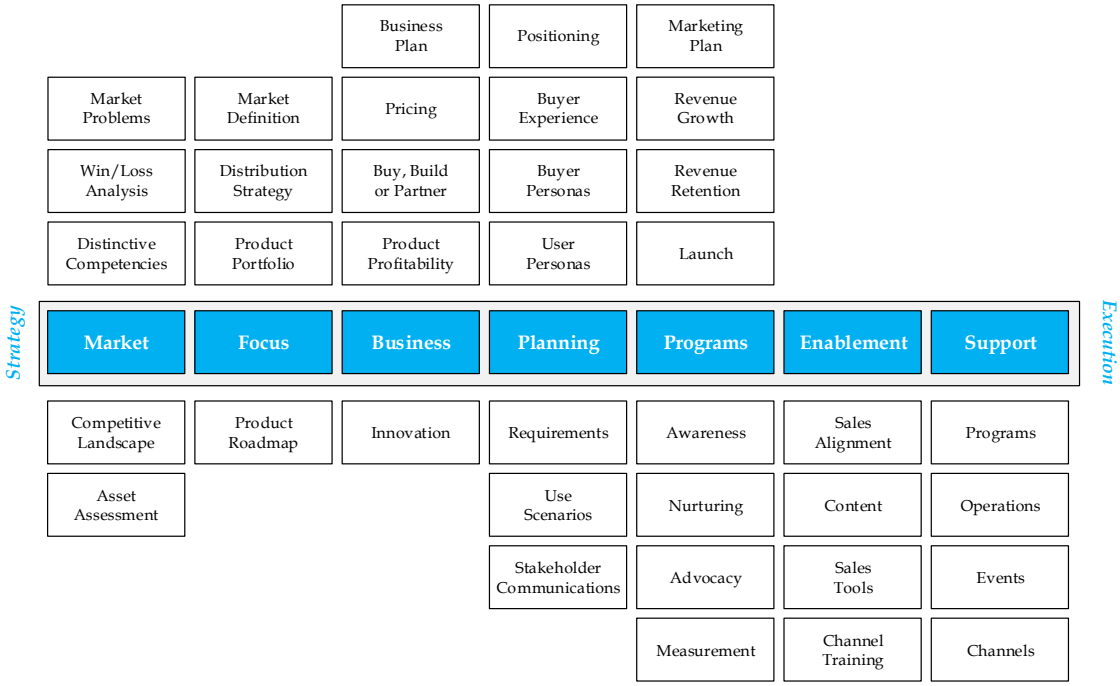

FIGURE: Pragmatic Framework (Adapted from Pragmatic Institute, 2019a)

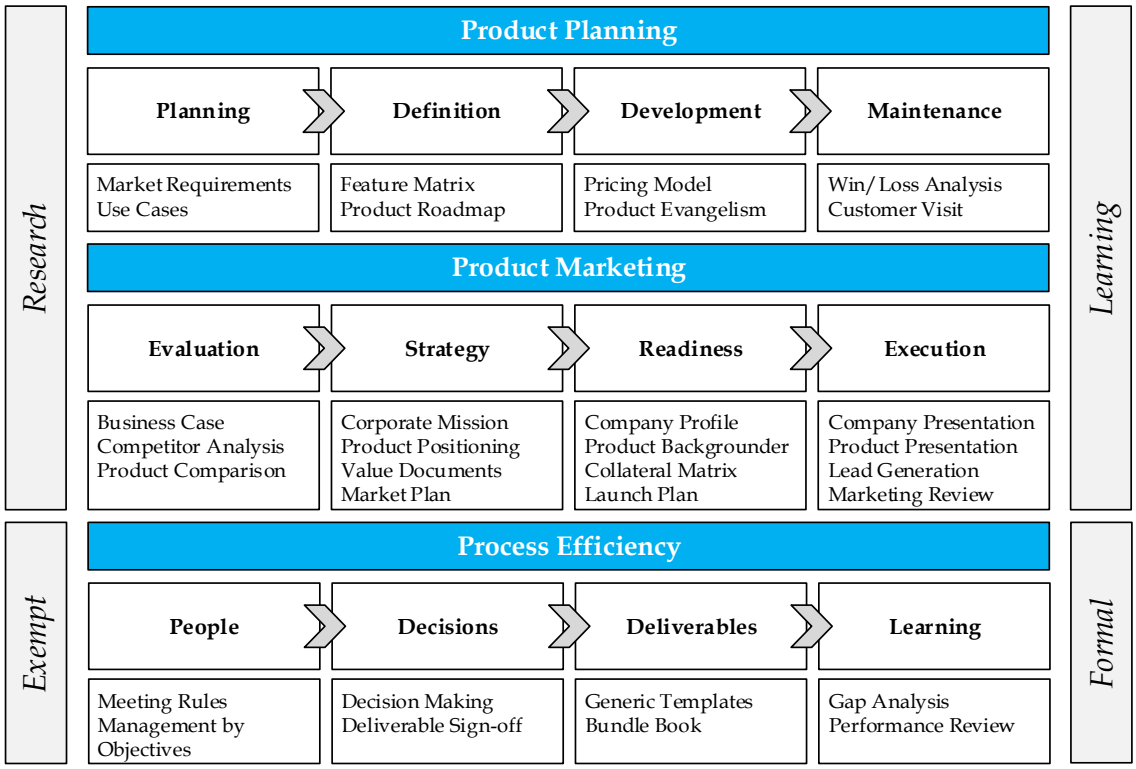

FIGURE: PMTK Action Model (Adapted from Blackblot, 2019g)

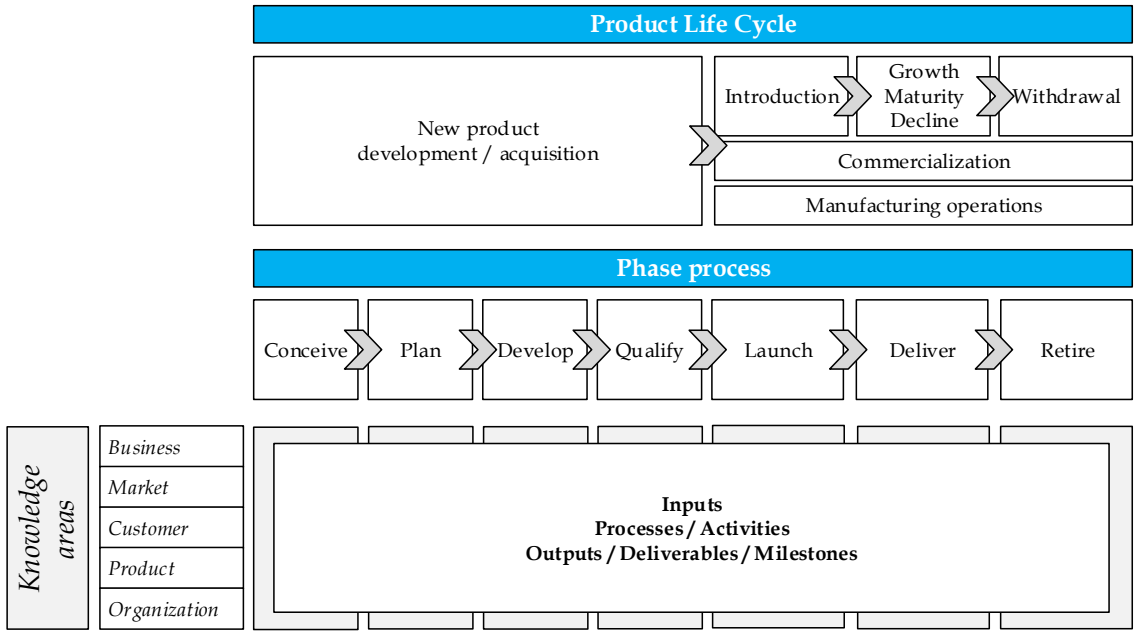

FIGURE: AIPMM / ProdBOK Framework (Adapted from AIPMM, 2017)

APPENDIX 2: SURVEY INVITATION

Invitation email:

From: *{ISPMA email list}*
To: *{recipient}*
Subject: Master's thesis research collaboration

Dear *{recipient}*,

In achieving organisation-wide benefits from the ISPMA SPM framework and the body of knowledge, the organizational structure and the processes related to software product management need to be implemented and/or improved.

But how in practice?

This question is of interest to ISPMA, and to a student from University of Jyväskylä (Finland), Antti Paajoki, who is researching the best approaches of implementing and improving practices based on ISPMA's SPM framework. Please help the student, and contribute with your expertise by answering the survey by Wednesday 18.3.2020.

The survey takes 5-15 minutes to complete, and can be found here:
{link to the survey}

In addition to contributing to the best practices in software product management, by completing the survey you will also have a possibility to win a 50% discount ticket to SPM Summit 2021 event (Frankfurt or Bangalore).

Thank you in advance for your help!

On behalf of the researcher,
ISPMA Board

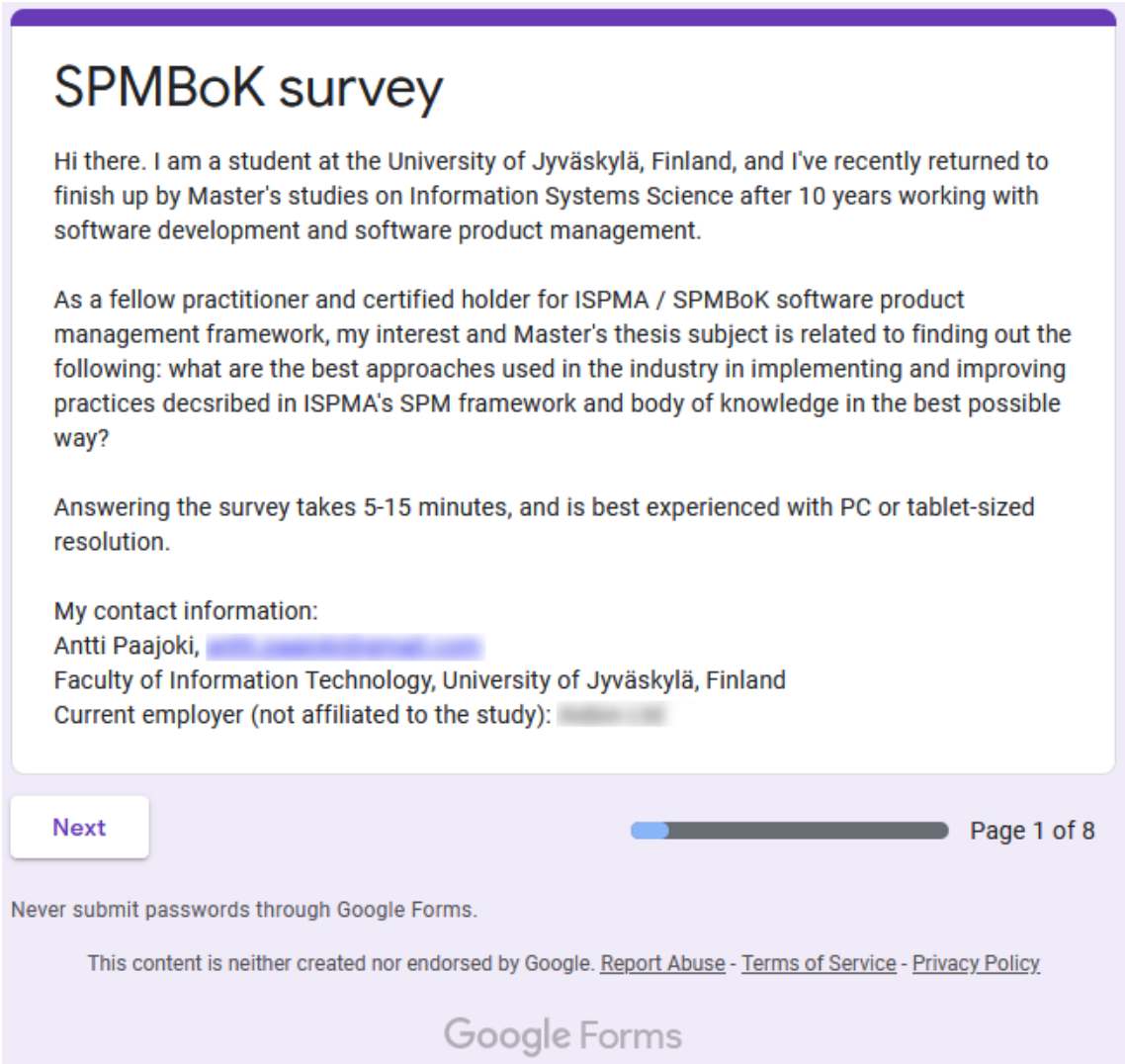

FIGURE: First page of the survey as shown in web browser

APPENDIX 3: STRUCTURE OF THE SURVEY

Section	Question	Type
Research notification	(Introductory text about the research)	
	I have obtained sufficient information about the study and the processing of my personal data. I have understood the information I have obtained and want to participate in the study.	Single-answer single choice (checkbox)
Personal background	(Introductory text about personal information)	
	Age	Single-answer multiple choice
	Gender	Single-answer multiple choice + Open answer
	Country (you live in)	Short answer
	Which of the following is closest to your job title?	Single-answer multiple choice + Open answer
	How would you classify the organisation you work in?	Single-answer multiple choice + Open answer
	Please select which ISPMA certificates you have.	Multi-answer multiple choice (checkboxes)
	Please describe if you have any other (software) product management related certificates.	Open answer
ISPMA's SPM Framework	(Introductory text and picture about the ISPMA's SPM framework)	
	Have practices based on the ISPMA's SPM Framework been implemented in your organisation?	Single-answer multiple choice
	Have you been personally involved in such process implementation or improvement?	Single-answer multiple choice
	When was the last time some or all of the processes were systematically implemented or improved based on the ISPMA's SPM Framework?	Single-answer multiple choice + Open answer
	Which other frameworks have you utilized to improve software product management processes?	Multi-answer multiple choice (checkboxes)
Success Factors	(Introductory text about success factors)	
	How important you see the following for the success of process improvements during a process improvement program?	Single-answer multiple choice question matrix (Likert-type scale)
	Based on your experience, are there other success factors than listed above?	Open answer
	Based on your experience on implementing practices based on ISPMA's SPM Framework, how satisfied you have been to the resulting processes after implementation or improvement in your company?	Single-answer multiple choice (Likert-type scale)
	Based on your experience on implementing prac-	Single-answer multiple

	tices based on ISPMA's SPM Framework, how satisfied you have been with the processes improvement process itself in your company?	choice (Likert-type scale)
	Based on your experience, what would help in implementing practices based on ISPMA's SPM Framework?	Open answer
Process Maturity	(Introductory text about process maturity levels)	
	How mature would you consider the following core process areas related to Software Product Management being in your company?	Single-answer multiple choice question matrix (Likert-type scale)
	Please select the five most important process areas, which you think your company should improve next.	Single-answer multiple choice question matrix (one per column 1…5)
	Please describe shortly why did you choose those process areas as the most important improvement areas?	Open answer
Company and product background	(Introductory text about success factors)	
	Size of your company	Single-answer multiple choice
	In which industry would you classify your company being in?	Single-answer multiple choice
	In which sector your company's customers primarily are in?	Single-answer multiple choice + Open answer
	Size of your company's product management function?	Single-answer multiple choice
	Size of your company's product function (all employees working on the product, including product management)?	Single-answer multiple choice
	How many colleagues in your company have an ISPMA certificate?	Single-answer multiple choice
	How many software products your company has?	Single-answer multiple choice
	How would you categorize your company's software products?	Single-answer multiple choice + Open answer
	What is a typical lifetime for your company's software products from inception to end-of-life?	Single-answer multiple choice
	How would you describe the level of customer involvement in the software management processes?	Single-answer multiple choice
	How would you describe the level of partner involvement in the software management processes?	Single-answer multiple choice
Summary	Would you like to participate in a lottery for discount ticket for SPM Summit 2021?	Single-answer multiple choice
	Would you be available for a follow-up video / phone conferencing interview regarding this research?	Single-answer multiple choice
	If you answered Yes to one or more questions above, please enter your email address.	Open answer
	Feedback	Open answer

APPENDIX 4: STRUCTURE OF THE INTERVIEWS

Background information

- How many employees does your company have?
- How have your company ended up in gaining ISPMA certification, and implementing/improving processes according to the ISPMA's SPM framework?
- Has the process improvement initiative been proposed by experts/specialists or by management?
- Have you used external help (consultants etc) in the process?
- How would you describe the usefulness of your ISPMA certification for the improvement process?

Theme 1: State of the ISPMA's SPM Framework implementation

- Is your company more in the process of initial implementation or improvement of software product management practices based on ISPMA's SPM framework?
- What are in your opinion the main drivers for the implementation effort?
- When were the initial processes introduced?
- Are the processes in use for all products or only for some?
- What do you think triggers the improvement needs?
- How often are improvements implemented into the processes?

Theme 2: Status assessment, guidance and tools for process improvement

- How have you assessed the current level of your software product management processes?
- Which kind of tools have been used to help the assessment?
- How have they worked, and have they been useful?
- Which kind of additional supporting tools, guides, or other help you think would be good for the assessment?
- How are you organized around this implementation/improvement process? Who are involved?

Theme 3: Ways of ensuring success in the process improvements

- One of the most important success factors for process improvement is management commitment. Could you describe what it means to you? Could you describe how it is or will be ensured in your company?

- One of the most important success factors for process improvement is staff involvement. Could you describe what it means to you? Could you describe how it is or will be ensured in your company?
- One of the most important success factors for process improvement is roles and responsibilities. Could you describe what it means to you? Could you describe how it is or will be ensured in your company?
- How would you describe the success of software product management process implementation in your company?
- What has been the most challenging in the software product management process implementation in your company?

Theme 4: Realized and expected benefits

- How would you describe the benefits of software product management process implementation in your company?
- Which are the most important measurable benefits you can think of?
- Which are the most important non-measurable benefits you can think of?
- Which benefits you would like to accomplish, but have not been yet able to do so?
- What do you think what could help you getting to those benefits?