International Journal of Business and Administrative Studies
volume 7 issue 2 pp. 89-106 doi: https://dx.doi.org/10.20469/ijbas.7.10004-2

Productization and Product Structure: Extending The Perspective to Software Business

Akwasi Gyamerah Adusei *
Industrial Engineering and Management,
University of Oulu, Oulu, Finland

Janne Härkönen
Industrial Engineering and Management,
University of Oulu, Oulu, Finland

Erno Mustonen
Industrial Engineering and Management,
University of Oulu, Oulu, Finland

Abstract: Productization and product structure can be beneficial for companies in various product-related considerations. The focus on software business and software products, however, has been deficient. This study takes a specific software perspective to provide a software business perspective to productization discussion. Productization of software is considered by bringing different perspectives on the software product. Constructive, qualitative approach is used in studying the productization and software product structure of a case company. The empirical study reveals software-related challenges and shows how productization and product structure are beneficial also with software business, including the Software as a Service (SaaS) business model. The results indicate that further focus on product structure and consistent productization logic is also necessary for software business. The basic principles of productization of physical products or services may remain the same, but some software business specifics must be addressed. Hence, software context may influence productization and product structure logic. Companies involved in the software business can benefit from the improved understanding in managing their software offerings. Well productized software offering enables maintaining the scope of the software being developed and the operations of what is being sold, delivered, and invoiced. This study is among the first to consider the productization concept, including product structure in the software business context to indicate the context specifics.

Keywords: *Productization, product lifecycle management, business and management, software business, SaaS, software product structure*

Received: 22 December 2020; ***Accepted:*** 03 February 2021; ***Published:*** 30 April 2021

INTRODUCTION

Responding to specific customer requirements through product differentiation is a reasonable means for companies to succeed in the global competition (Barney, Aurum, & Wohlin, 2008; Pandla, 2016; Ratnayake & Markeset, 2010). However, in meeting these demands, companies tend to face increasing pressures due to rapidly changing technologies and shorter product lifecycles; hence, effective resource allocation is emphasized for their survival and prosperity (Cooper, Edgett, & Kleinschmidt, 2001). Companies need to enhance the possibilities to quickly understand the customers, company products, product cost, product quality, and overall profitability (Sakti, 2016; Lahtinen, Mustonen, & Harkonen, 2019). This will go a long way to deal with prevailing consequences such as product confusion among

*Correspondence concerning this article should be addressed to Akwasi Gyamerah Adusei, Industrial Engineering and Management, University of Oulu, Oulu, Finland. E-mail: akwasi.adusei@oulu.fi customers (ElMaraghy et al., 2013), product portfolio explosion (Tolonen, Harkonen, Haapasalo, et al., 2014), and product cannibalization (Srinivasan, Ramakrishnan, & Grasman, 2005). Productization is one way to address products and focus on the combinations of product elements, regardless of the nature of the product (Harkonen, Haapasalo, & Hanninen, 2015). However, the focus on productization has been deficient in the software product context.

Productization is defined as "the process of analyzing a need, defining and combining suitable elements, into a product-like object, which is standardized, repeatable and comprehendible" (Harkonen et al., 2015). Productization is seen to have a process nature and involves analyzing customer needs while satisfying these needs (Flamholtz, 1995). Suominen, Kantola, and Tuominen (2009) view productization as a standardized process from the early product considerations towards commercialization. However, the current productization discussion in the software context is relatively narrow, with vague linkages to the software development stage (Kim, Kim, & Kim, 2007; Rautiainen, Vuornos, & Lassenius, 2003; Wallin, Ekdahl, & Larsson, 2002). The discussion in the software context needs strengthening.

The product structure considerations are also linked to productization (Hannila, Tolonen, Harkonen, & Haapasalo, 2019; Lahtinen et al., 2019; Mustonen, Harkonen, & Haapasalo, 2019). From the software perspective, productization has been viewed as an emerging research area (Mathur, 2006) where the discussion links to areas such as software product management (Helferich, Schmid, & Herzwurm, 2006), delivering of standardized software products (Parry, Jones, Rowley, & Kupiec-Teahan, 2012; Vlaanderen, Brinkkemper, & van de Weerd, 2012), and software product value demonstration (Feller, Finnegan, Fitzgerald, & Hayes, 2008). However, productization through product structure has not been discussed explicitly in the software context. The discussion merely has indications of software forming a part of the product. Overall, the productization activities relate to creating commercial readiness to enable selling, delivering, using, and invoicing (Harkonen, Tolonen, & Haapasalo, 2017, 2018a, 2018b). However, the previous research has not studied similar aspects of software business specifically. Most literature on product structure in conjunction with productization has focused on physical products (Harkonen et al., 2015) or services (Kuula, Haapasalo, & Tolonen, 2018), not so much on software. Different forms of selling software products, such as SaaS, also necessitate further clarification in the context.

Research Gap

Productization of software concerning product structure has been recognized as very important in the software business, but its discussion is limited in the extant literature. The software business perspective has been deficient in the literature in terms of productization discussion. Also, the different forms of software products have not been considered in this context.

Novel Contribution

Contribution is provided by highlighting the significance of understanding the software business offerings as products and that it is the productization that defines the outcome. It is pointed how productization inherently structures the software product, provides a basis for systematization, and reference points for data.

Research Significance

Productization and considering the specific structural nature of software products enable the management of the software products better and benefit the commercialization through the created capabilities. The significance of this study lies in considering different forms of software product delivery alongside productization logic and the product structure. As a result, new insights are provided into software context specifics.

Research Objectives

This study focuses on software products and productization to provide a software business perspective to productization discussion. Three ways of productizing for software business are considered, including SaaS. The aim is to broaden the understanding of the productization of software products. The study takes a constructive research approach and reviews the literature and the empirical observations in the software company context.

Research Questions

The following research questions are devised to support the objectives:

RQ1: What is the current state of productization and product structure in the software context?

RQ2: How should the product structure and productization support software products and the software business?

LITERATURE REVIEW

Software Product

The product can be defined as physical, software, service, or any combination (Hannila et al., 2019). Various researchers have provided definitions depending on the context or scope (Peltonen, 2000). According to Haines (2014) and Ulrich and Eppinger (2012), anything that can be sold to a customer, or an enterprise, be it goods, services, or knowledge, is a product. A company's product enables to offer solutions to the customers and therefore provides the primary resource pull on which the company survives (Kropsu-Vehkaperä, 2012; Stark, 2016). Products are often discussed as either tangible or intangible (Kahn, 2012; Stark, 2016). Software-based offerings, including computer programs, procedures, documentation, and data delivery to users and related service offerings, are referred to as intangible products (S. A. Fricker, 2012). Software products can be seen to link to digitalization that corresponds to digital technologies and new ways of value creation (Gbadegeshin, 2019). Digital products can involve software (S. Fricker & Kittlaus, 2017), platforms (Cusumano, 2019), or digital services. Harkonen et al. (2015) state that products can be defined as the suitable combination of necessary elements that constitute an offering that can be sold to customers to satisfy their needs. Customers judge each product in the market on three essential aspects, product features, quality, product service mix, quality, and product price (Kotler & Keller, 2012).

Literature has attempted to shed light on software products, especially in the software business context. For instance, Xu and Brinkkemper (2007) discussed software products as packaged configurations of software components or software-based services with auxiliary materials released for and traded in a specific market. By definition, the software is more than computer programming files, consisting of associated procedures, documentation, and data (Gharbi, Koschel, & Rausch, 2019). In a literature review on engineering requirements for software product lines (V. Alves, Niu, Alves, & Valença, 2010), software product lines were explained as "a set of software-intensive systems sharing a common, managed sets of features that satisfy the specific needs of a particular market segment or mission, and that are developed from a common set of core assets in a prescribed way." Other software related topics such as packaged software (Light & Papazafeiropoulou, 2004), Tailored software (Hietala, Kontio, Jokinen, & Pyysiainen, 2004; Sawyer, 2000), Commercial-Off-The-Shelf (COTS) (C. Alves, Pereira, & Castro, 2006), among others have been discussed by the literature to explain the software as a product concerning software business. Nevertheless, software products involve a certain malleability (possible to re-program), homogeneity (use of standard language), and transferability (the ease of transferring a digital object), which influence their nature (Hinings, Gegenhuber, & Greenwood, 2018). The concept of software productization has received attention in the software business because software companies have identified the need to change from one customer-specific software delivery, popularly known as tailor-specific, to a more standard software that could serve a target market (Artz, Van De Weerd, & Brinkkemper, 2010). However, the research is still limited in this subject as researchers and companies have yet to cover a logical entity regarding business models, development, and software implementation. Examples of topics that have been generally discussed include requirements management, configuration, and delivery (Xu & Brinkkemper, 2007).

SaaS has received worldwide attention from most software industries (Dubey & Wagle, 2007; Jayasimha, Nargundkar, et al., 2020). Unlike the traditional model where software vendors sell a permanent user license to their customers, SaaS has to sell subscriptions to users (Choudhary, 2007). The conventional way where customers were challenged to have all the technical expertise, as well as high capital to be able to manage their purchased software, is now seen to be taken care of by the SaaS model due to its web-hosted services (Wu, Garg, & Buyya, 2011). So defined, the SaaS delivery package consists of a cloud-based software application, fully managed by software sellers, and can be accessed by customers over the internet based on a subscription fee (Bhardwaj, Jain, & Jain, 2010; Choudhary, 2007). In general, cloud-based software is classified to include: SaaS, Platform as a Service (PaaS), and Infrastructure as a Service (IaaS); however, SaaS is the only cloud-based service that offers similar solutions as traditional on-premises packages (Godse & Mulik, 2009). One of the critical factors for any successful SaaS company is the configurability of the SaaS products' configurability enables the customers to choose from the list of options and variants to have their software configurations (MCSI, 2009). Examples of business applications that SaaS adopts include sales force automation such as customer relationship management, e-commerce, and payroll.

Productization

Productization has a role in forming the basis for effective product management, product portfolio management, and product data management (Hannila et al., 2019; Harkonen, Mustonen, & Hannila, 2019; Lahtinen et al., 2019; Tolonen et al., 2014; Tolonen, Harkonen, Verkasalo, & Haapasalo, 2015a). The consistent logic for dealing with products is possible via productization (Harkonen et al., 2018a). This logic is linked to product structure and lifecycle considerations (Lahtinen et al., 2019). Pyron, Prado, and Golab (1998) define productization as including the activities required to achieve a product's commercial readiness. However, productization is deficiently studied in the software context. There are still some grey areas relating to the software product structure and productization, especially concerning portfolio management in the software business.

Alajoutsijärvi, Mannermaa, and Tikkanen (2000) stated that to achieve overall continuous success in the software business, there is a shift from peculiar customer projects towards actual standardized products, productization is always a prerequisite. Software productization can be defined as "the process of converting routine software functions into modules which are used as building blocks for various applications" (Youngdahl, Ramaswamy, & Dash, 2010). Rautiainen et al. (2003) relate productization in the software development context as a product release process where all stakeholders contribute. In software productization, different characteristics are recognized in the discussion after noticing technology-related inconsistencies. The characteristics include; improving customer understanding, demonstrating value, tangibilisation, standardization, and reproducibility. Artz et al. (2010) discussed software productization to address the characteristics of standardization. The description by Artz et al. (2010) focused on productization as a process that changes the software offering from customer-specific software to a standardized software, whereby the process involves six stages from independent projects, project features reused, product recognition, product platform, standardized product platforms, and lastly customizable/standardized products. However, software productization is still lacking when it comes to customer understanding enhancement, tangibilisation, and value demonstration. Software context-specific focus on services is also limited, but in general, the productization of services is understood to translate the abstract service and its creation into concrete exchangeable objects and controllable processes (Jaakkola, 2011).

Software Product Structure

In general, product structure represents the product components, properties, and relationships between them (Crnkovic, Asklund, & Dahlqvist, 2003). The product structure can create a general understanding of a company's products and offerings among different stakeholders. It helps companies manage product variants and makes product data management more efficient (Kropsu-Vehkaperä, 2012). A product structure supports understanding product hierarchies (Crnkovic et al., 2003). Further, the product data and the relationship between the product elements are encapsulated in its product structure (Saaksvuori & Immonen, 2008; H. Zhang & Jarzabek, 2004). A standard product data model may also be necessary to efficiently manage product data in information systems (McKay, Erens, & Bloor, 1996; Sudarsan, Fenves, Sriram, & Wang, 2005; Svensson & Malmqvist, 2002). The digital transformation is likely to also influence the software business due to all sectors increasingly relying on data. The human centricity of new technologies will change the focus (Dang, Amin, Shihada, & Alouini, 2020), further emphasizing the ability to address the data consistently. However, the previous literature does not specifically discuss the software product structures in the context of productization.

Harkonen et al. (2017) were among the first to present the product structure model in the productization context to consist of two sides: the commercial product structure and the technical product structure. The commercial product structure shows different top-down levels, including solution level, product family levels, product configuration levels, and the sales items level visible to the customers. At the same time, the technical product structure also consists of different levels of version items, assembly, and sub-assembly, which are typically visible only to the company. The previous works indicate applicability for software products but fail to provide deeper discussion in the software business context. Also, Lahtinen et al. (2019) further elaborated on the model over the lifecycle. Again, however, the software business context necessitates further discussion.

Most subjects and terminologies discussed by the literature in the software product structure context are geared toward software development processes and methodologies. Some terminologies discussed by the literature include agile methodologies (Kumar & Bhatia, 2012; Schwaber & Beedle, 2002; Cockburn, 2004; Subair, 2014), software architecture (Aldrich, Chambers, & Notkin, 2002; Solms, 2012), structure-oriented programming and object-oriented programming (Asagba & Ogheneovo, 2008). Nevertheless, the software product structure may also involve service processes and be linked to resources and competencies (Kuula et al., 2018).

Software Configurability and Modularity

Configurable products consist of a set of components that are well defined and have predefined interactions with each other (Sabin & Weigel, 1998). Product configurability enables the final product to be created according to customer choice by using predefined elements. For example, a variant provides a group of alternatives for the customer to select, while a product option provides the characteristics that are either selected or left out (Kropsu-Vehkaperä, 2012). Configurable product definition includes specifying these variants and options, as well as the necessary rules and constraints. A new product version replaces the previous older one while product variants extend the product family (ElMaraghy et al., 2013).

Amidst the increasing attention researchers have placed on configurability, most studies on configurability have been done on physical products. Literature has not discussed enough the configurability in the context of the software domain (Asikainen, Soininen, & Männistö, 2003). Product configuration is characterized by a modular structure whose individual products are specified by predefined components where the combination follows well-defined rules. With software product configuration in place, Asikainen et al. (2003) highlighted that software companies do not start from scratch by designing different products for each customer through specific coding. However, customer requirements are configured through the selection of components and other parameters. However, S. Zhang, Shen, and Ghenniwa (2004) study shows large variability in software product lines because of customer-specific needs, system maintenance, and other environmental factors. Therefore, there is the possibility for the explosion of variant combinations resulting in difficulties in variant configurations. They added that to tackle these problems, there is the need to design software languages that support variant automated customization. A configurator provides a platform that guides the choice of customers based on the customers' requirements. It aligns the requirements with choices compatible with the configuration model (Asikainen et al., 2003). Software productization through product structure captures the product variants, which are very useful for product management (Kropsu-Vehkaperä, 2012).

On the other hand, modularity provides the means to increase the commonality across different product variants within the product family. Similar product components are used in multiple product variants and all product variants (Salvador, Forza, & Rungtusanatham, 2002). According to (ElMaraghy et al., 2013), product modularity provides a condition where the components may be separated and recombined to form a new product variant. Modularity can also be considered in the case of software.

In literature, the topic of modularity concerning software has been broadly defined and understood depending on the context of the discussion, and therefore a common definition has still not been agreed upon by researchers (Kästner, Apel, & Ostermann, 2011; Ostermann, Giarrusso, Kästner, & Rendel, 2011). In Ostermann et al. (2011) discussion on the road to feature modularity, "Modularity denotes the degree to which a system is composed of independent parts, whereby 'independent' implies reusability, separate understandability." Any well-defined boundaries and components of a software product, either as a tool, application, or operating system, can also be classified as modular products (Meyer & Webb, 2005). The subject of software modularity can be described in software under two notions: cohesion and information hiding without and with the interface, respectively (Kästner et al., 2011), all of which have a different modularity definition. However, a good software architecture enhances modularity among the software components, thereby making the possibility to recombine the components to achieve different needs without the need for additional codes (Meyer & Webb, 2005).

Literature Synthesis

It is essential to understand that anything that can be sold to customers as goods, services, software, or knowledge can be considered products. Software-based offerings, including computer programs, procedures, documentation, and data delivery to users and service offerings, are referred to as intangible products. Software, too, should be seen as a product. Software products entail packaged configurations of software components, or software-based services, with auxiliary materials released to and traded in a specific market.

The SaaS business model has received much attention from the software industry. SaaS consists of cloud-based software applications managed by software vendors where customers can access through the web browser. SaaS services are mainly paid by customers through a subscription fee, unlike permanent user license fees paid for by customers during traditional software purchases. Configurability is also a very beneficial aspect of selling SaaS products. Productization through the product structure can organize the SaaS product in a logical relationship and hierarchy helpful in the configurator.

Productization creates a logical consistency for dealing with products, and this logic is linked to product structure. In software productization, routine software functions are converted into modules used as building blocks for various applications. Software productization also creates the possibility for stakeholders to make their contributions, especially during software development. Software productization can be characterized by improving customer understanding, demonstrating value, tangibilisation, standardization, and reproducibility.

Generally, the product structure presented in both the commercial and technical structure is an essential backbone in the productization concept. It shows the product components, properties, and the relationship between them. It serves as the vehicle for creating an understanding of the offering, managing product variants, and facilitating product data management. The software business's product structure and productization concept are mainly geared towards the software development approach, where discussions are around software architecture and agile methodologies. A commercial and technical representation of software product structure is not widely discussed by literature, and this gap is mobilized to consider how software business can be productized. Software configuration, modularisation, and effective product portfolio management are all complementary perspectives facilitated through productization and the product structure. Figure 1 synthesized the essence of software productization and product structure.

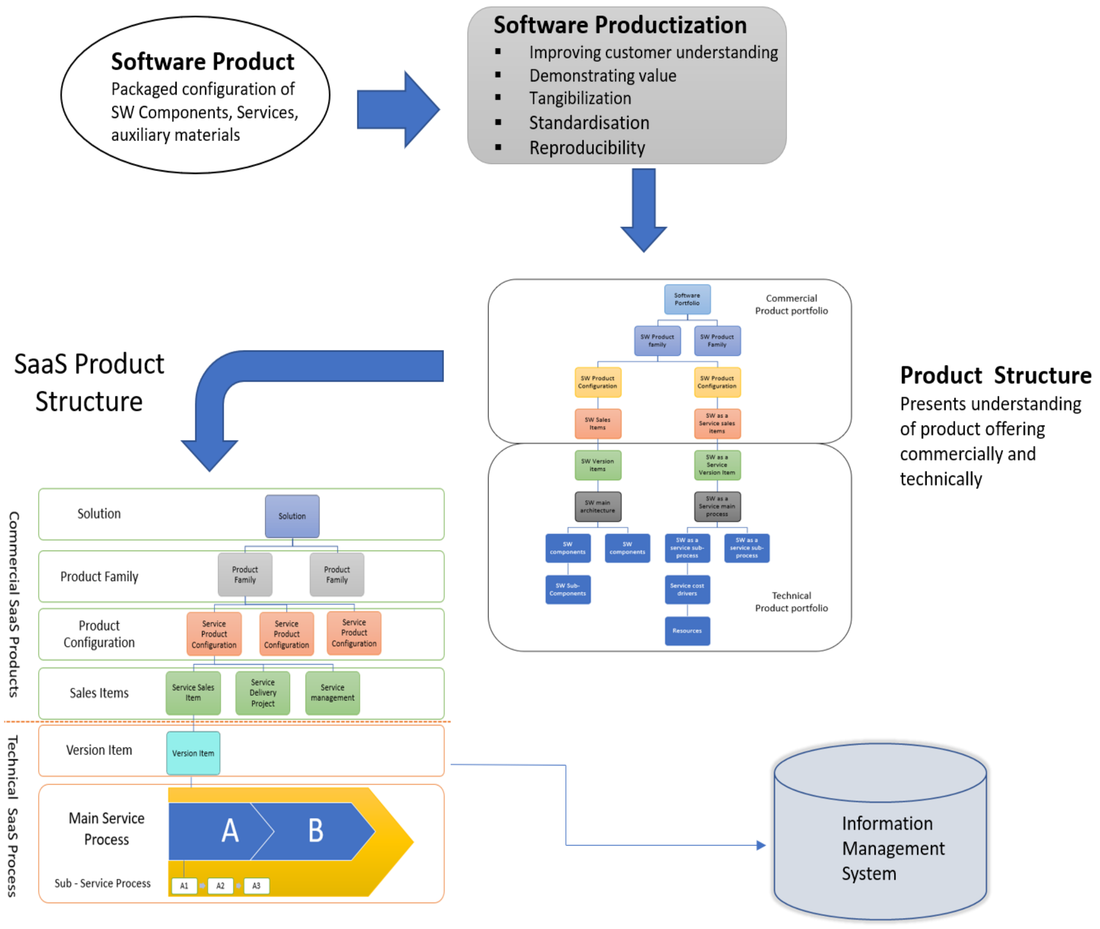

Figure 1 *Software Productization and Product Structure [Author original]*

RESEARCH PROCESS

Research Design

Research design, the research structure guide the realization of research methodology and the analysis but mostly answers the research questions (Saunders, Lewis, & Thornhill, 2007). This study takes a somewhat descriptive approach as opposed to a purely theoretical one. However, the approach is somewhat middle ground in synthesizing an explanatory framework based on prior literature to deal with the essence of productization and product structure in the software business context. Literature hence directs the analysis of empirically collected data. The research strategy involves inductive reasoning and qualitative data as a manner of investigation. The chosen research design involves constructive research. In constructive research, a research problem is attempted to address by building new construction (Kasanen, Lukka, & Siitonen, 1993). The comprehension of productization and related product structure is supported by creating a construction. The research methods are the means and the more used for collecting data.

Research Method

The research is structured as illustrated in Figure 2. The study is qualitative and seeks to address productization and product structure software specifically. The constructive research approach is applied (Kasanen et al., 1993). An extensive literature review forms the foundations for the study. The literature-based understanding was further utilized to support the empirical investigation.

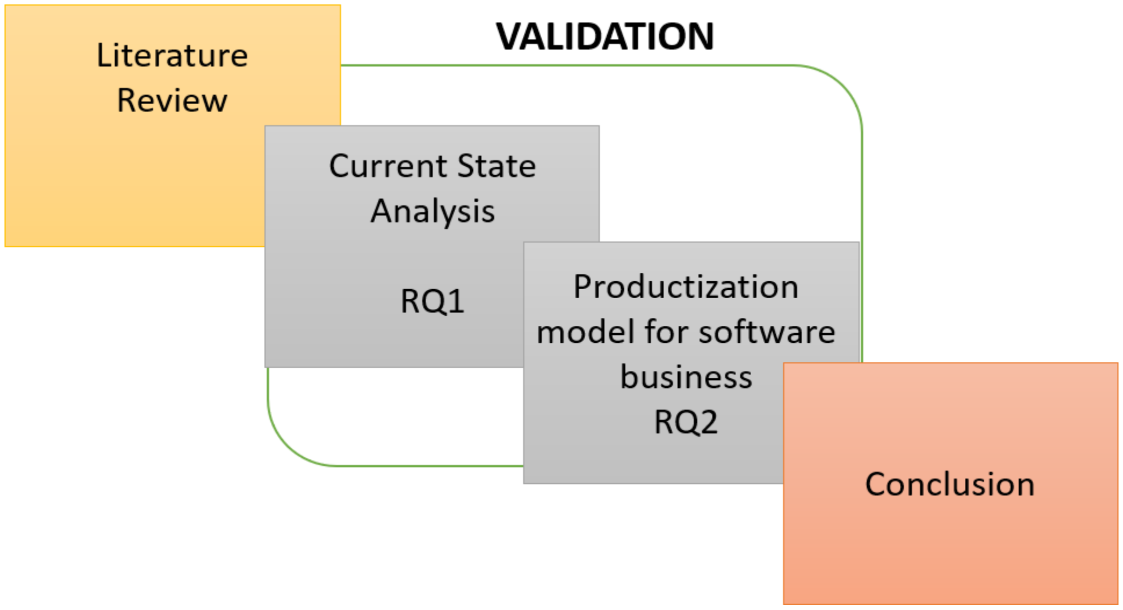

Figure 2 *Research Process [Author original]*

Empirical data was collected through semi-structured interviews (Merton, Fiske, & Kendall, 1990). The interviewees could discuss the topics as entities. The specific focus was on addressing the productization logic and the structure of the software offering. The study focuses on analyzing company practices based on the possibility of having access and company interests in advancing the study area in the business context. The population that this study concludes about involves those company employees linked to relevant company practices and the products commercially or technically in any of the company locations internationally. Snowball sampling was applied to identify knowledgeable informants (Biernacki & Waldorf, 1981). Company employees were interviewed widely to cover all the key roles. Some informants were interviewed more than once as they proved to be extremely relevant. The interviews covered the personnel with representation from different related areas. Different perspectives were intentionally included to reduce the bias. Based on the criteria of experience, position, and in-depth knowledge of company products and practices, each interviewee was carefully selected to obtain the best and accurate empirical data to be analyzed. Eleven key persons were interviewed separately where each interview session lasted for two and half hours on average. The interviews helped to obtain different individual views and knowledge about the current productization and product structure. Table 1 indicates the different departments and roles of the interviewed people. Also, wide access was provided to company internal materials and all the documentation relevant to the software offering. This allowed triangulation. Publicly available material was also utilized.

One of the researchers spent several months at the company premises and had numerous additional discussions with the employees. The analysis took place over several months. Apart from the general study of the company products, a specific product offering was chosen and studied more carefully. Finally, the findings were distilled, and a model for productization in SW business was proposed. The model is based on the gaps between the literature and the current state of productization and product structure. Separate feedback and review sessions were organized to validate and finetune the model.

Table 1 *INTERVIEW PARTICIPANTS AND THEIR ROLES*

Department	Role
Product management	Head of Product Management
Product management	Product manager
Product management	Product manager
R&D	Head of R&D
R&D	Chief Software Architect
Customer and Sales	Head of Customer and Finance
Customer and Sales	Chief Specialist
Customer and Sales	Sales Director
Project Management	Head of Customer Projects
Project Management	Project manager
Customer Support	Specialist

Company Description

The analyzed company works in the service business area and operates in different business sectors. They provide solutions through a digital platform in power plants and factories maintenance, smart grids and telecommunications connections, modular software development, and services for big data management for energy customers. The company has different digital products that provide different solutions in the energy sector, achieve production optimization, and health and occupational safety. Therefore, the energy business sector was chosen as the focus of this study.

In the energy business sector, the company provides B2B modern and intelligent information management service solutions to the entire energy market value chain to improve business efficiency. The main customer segments include energy production, electricity transmission, and energy sales companies. It offers solutions through the services and digital platforms to different customer processes focused on producing, distributing, buying, selling, or trading energy.

Their main offering comprises 1) ICT-based system solutions, including the meter data management system, energy management system, customer information and billing system, information and exchange services, and site management. 2) Operational service solutions include electricity procurement planning, balance settlement, energy consumption, and procurement forecasts, trading, and wind power services and 3) Control room services which act as the extended arm of its customers, who deliver monitoring of production plants, transformer stations, and supply system.

The studied case product is a high-level automated service system designed to help energy companies manage their customer information and invoicing. It provides different functions for sales and distribution of electricity, district heating, and so on. Also, it has a log function for the monitoring and analyzing of information security. In addition, the product platform provides flexibility where customers can customize a solution that meets their business needs.

RESULTS/FINDINGS

The Current State of Productization

The empirical study revealed that the productization practices had deficiencies in the Software (SW) context and how their relevance was not fully understood. It became evident that the consistency of software product items was a major challenge making understanding the product logic challenge. The interviewees indicated that certain product terminologies such as product solution, product family, product configurations, and versions are known and used to a certain extent, but certain variations exist in how the terminologies are understood. In defining the basic concept, several understandings existed: According to the first interviewee, "A product is our service business lines and digital platforms." The second interviewee defined it as "A product is something that is sold to our customers," and the third interviewee said, "I don't know; it's our work that brings us money." This at the initial stage suggested a significant level of inconsistencies in understanding, which challenged the effective communication among stakeholders. The company knows the market segment of the analysed case product, and no documentation was deficient except the productization practices. This deficiency has enabled each employee to form their own opinion regarding the company's offering. For instance, it was recorded that some people see the case product as a product family, while others see it as a configurable product that is sold to customers. The different views about the company's sales items indicated the lack of clearly defined sales items.

Also, the possible software product configurations and the sellable items had not been considered adequately. As a result, there was no clear picture of the product offering. This was indicated as a challenge for sales and created confusion among the customers, and even the company functions, including the sales team, product management, business management, support, and customer projects team. The general practices in the company appeared to base on customer-specific projects, with an approach other than a product-based approach. However, the company had a clear aim of shifting towards a more product-like delivery. Therefore, it could be concluded that the productization was deficient.

Software productization through the product structure consideration was lacking and made it challenging to move towards product standardization. The current situation regarding the structure of the case product is such that it is viewed mainly from the software architecture perspective only, although it also operates as a service type of product. The commercial and technical structures of the case product were not well defined. A high-level technical presentation and technical structure in the development environment, specifications, and architecture drawings exist; however, description of software versions, software service processes were highly deficient in the documentation.

By considering the product structure model and carefully studying the software product in focus, it was realized that the software product is known to satisfy customer needs through three main service offerings. These include the customer's own premises package, SaaS package, and Business Process as a Service (BPaaS) delivery package. However, the offering had not been productized clearly, nor had the logic been consistent among the products. Therefore, new product structures were devised and proposed under the three service offerings.

Productization Model for SW Business

The product structure for each of the offerings is recommended to have a commercial and a technical part. The commercial product structure for each model is proposed to have different product structure levels, including solution, product family, sub-product family, product configuration, and individual sales item levels. The commercial product structure is visible to the customers and consists of items that can be ordered, delivered, and invoiced. The technical product structure for each service model is also proposed to consist of various levels, including version items, main processes, sub-processes, and tasks. Each sales item is linked to its corresponding version items, showing all the version changes technically involved in each of the sales items. The individual service processes, sub-processes, and tasks can be linked to the needed resources through cost drivers to calculate service costs. Instead of the traditional view of seeing software as an entity having its software structure, the software is now seen as a resource in service production.

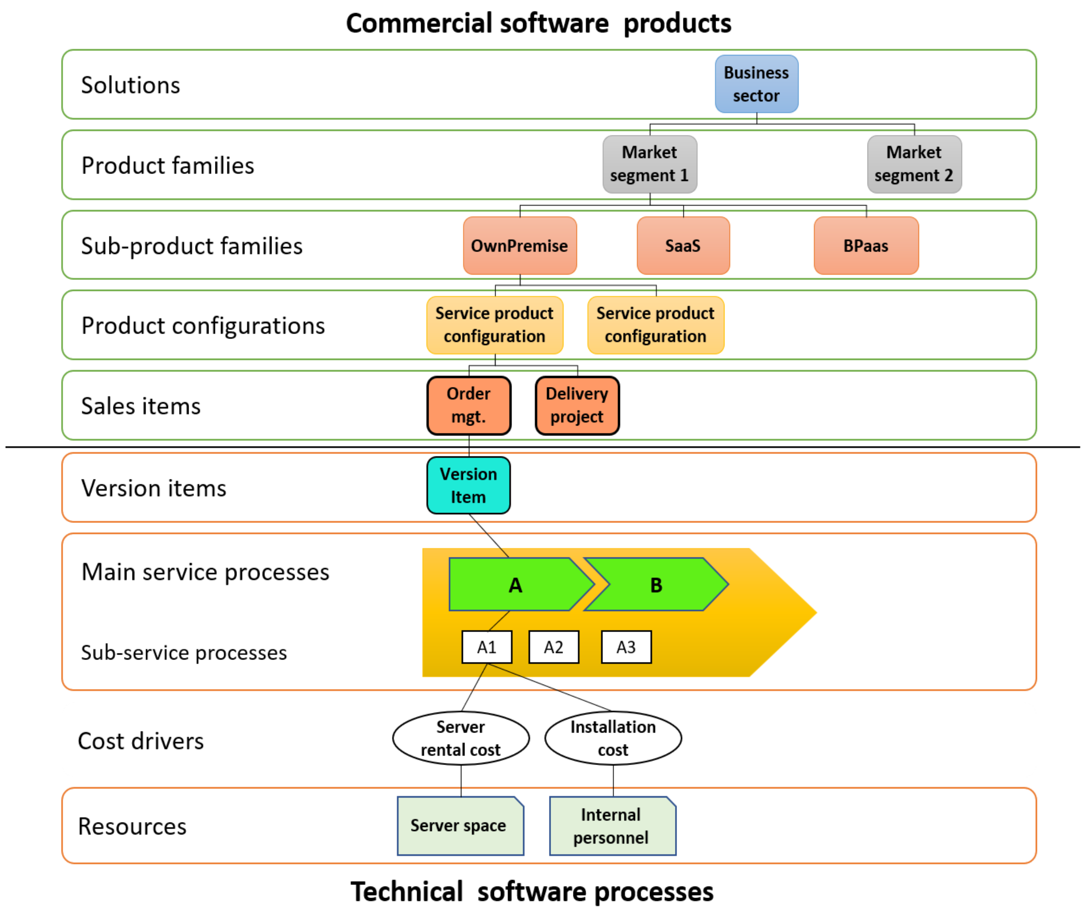

Figure 3 *Construction of Own Premise Service Offering Product Structure [Author original]*

Customer's premise package: This offering involves setting up custom software configuration on their servers in the customer's premises. The customer then operates the software and maintains the infrastructure by themselves. The analyzed company provides only technical support and sells new features and releases for the software.

Figure 3 shows the recommended product structure for the customer's own premises service offering. The solution level constitutes the business sector under consideration. The business sector is sub-divided into different market segments, which the case product will service. Each market segment represents a separate product family in the product family category. Below the product family is the sub-product family level. The Own Premise services are applied using the case product under the different market segments. It is important to note that the requirements for each market segment are different, and therefore the case product provides different Own Premise services for them. The next level after the sub-product family is the product configuration level. Customer-specific service products are configured here by sales items. The product configuration is only possible through the list of the sellable items at the sales items level. The sales items level is the last stage in the commercial Own Premise product structure, and at this stage, customers have the freedom to make their own choices from the sellable items to configure their service product based on their business needs. It should be noted that Operation service and Infrastructure as a Service (IaaS) are not listed among the sellable items for the Own Premise package since installation and all operations are done in the customer's servers and by the customers, respectively. The detailed levels in the technical Own Premise product structure include version items, main processes, cost drivers, and the resources needed.

SaaS package: In the SaaS, the analyzed company installs a software configuration in their server farms while all the customer's operations, such as invoicing, balance settlement, and metering services, are done by the customers. The major role of the case company is to ensure good level maintenance (updates and upgrades), database, hardware environments, and smooth running of their servers. In addition, they also provide technical support to the customers.

In Figure 4, a suggested product structure for SaaS service offering is shown. The solution level shows the targeted business sector in which the service product is going to serve. The solution level contains the different product families. In the SaaS product structure, each market segment represents one product family where the case product can provide different services. Under the individual product, families are the division of sub-product families. The SaaS services applied by the case software A at each market segment are in the sub-product family level. Customers can now configure their SaaS service products at the product configuration level depending on their business needs. The last level in the SaaS commercial product structure is the sales items level. Customers select their preferred sellable items that are needed to configure their service products. Again, SaaS sellable items exclude the operation services since customers choose to perform their operations. This is the major difference between BPaaS and SaaS. Technical SaaS product structure shows all the version items, main processes, sub-processes, cost drivers, and resources for each sellable item.

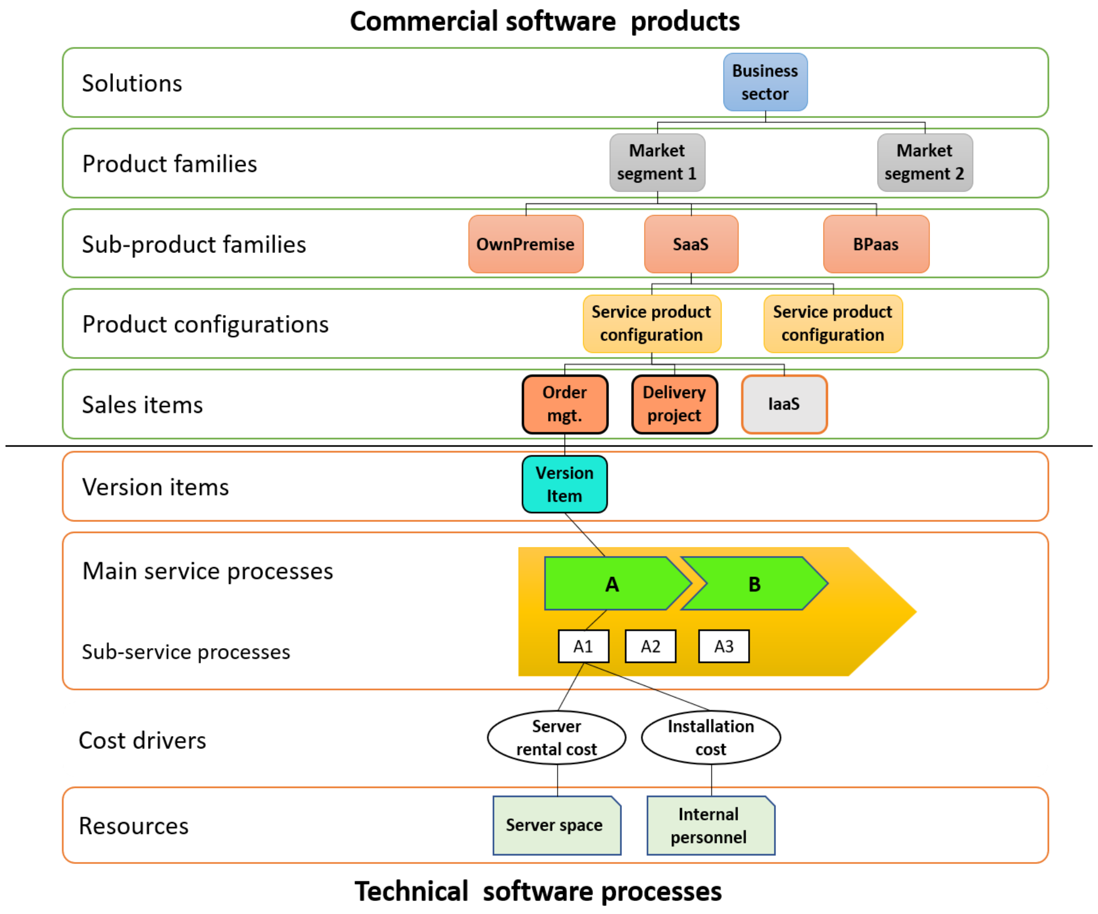

Figure 4 *Construction of SaaS Service Offering Product Structure [Author original]*

BPaaS package: The product is only sold as a managed service package in this service offering, and the customers only pay for the services. In this package, the analyzed company uses its tools, does the installation in its server farms, and carries out the customer's business process operation. The end customers can access this service delivery through a web browser. In other words, the case company is sub-contracted to provide the service operations on behalf of their customers to the end-users.

Figure 5 shows the product structure for the BPaaS service offering. The proposed solution level, which represents the highest level of the hierarchy, is the business sector. Below the solution level is the product family level, which shows the family of different market segments served by the case software A. The product family level comprises the sub-product family levels where the product offers different service models, including BPaaS. Under the sub-product family level is the product configuration level. With the product configuration level, the customer business process services are configured. Each configurable service product is linked to individual sales items where customers make their own choices for their configuration. The sales items level represents the last level in the commercial product structure, and the sales items include both variant sales items and optional, add-on sales items. It is up to customers to make their choices to be configured to meet their own needs.

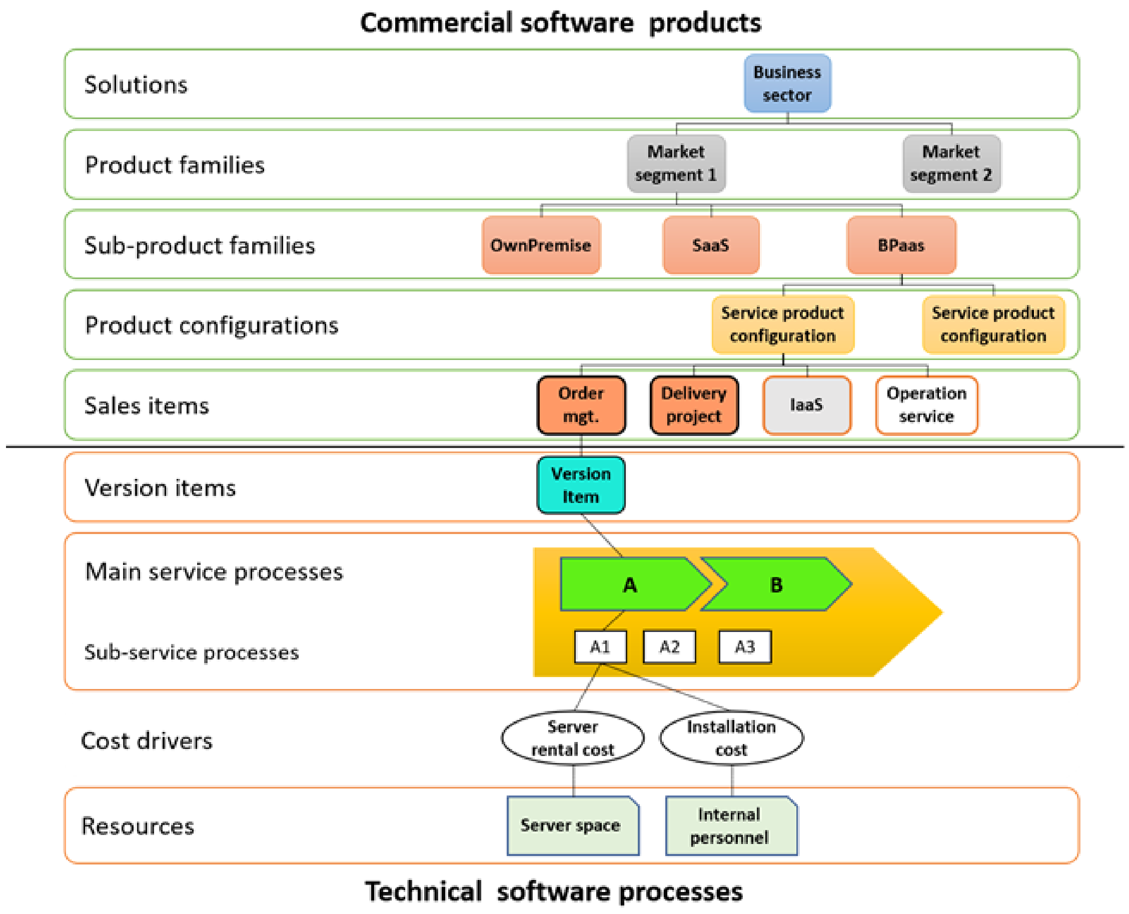

Figure 5 *Construction of BPaaS Service Offering Product Structure [Author original]*

The technical product structure shows each version item, main processes, sub-processes, cost drivers, and resources. A similar product structure is repeated for BPaaS in other market segments. In other words, the product structure showing BpaaS services provided by the case product is tailored to suit the distinct need of the other market segments.

It must be noted that software businesses should define their product offering in a very clear and consistent logic to promote internal and external stakeholder understanding, tangibilisation, and software product standardization. Commercial and technical productization makes it possible for customers and software developers to better view software products. Providing a software service offering in the software business also requires productization through a product structure, as individual service package needs to be clear enough for effective interaction between customers and software companies. The results show that the software service industry can serve customers' business needs through three main service offerings: the BPaaS, SaaS, and customer own premise, and it is proposed to have a product structure under each service package.

DISCUSSION

This study aimed to extend the understanding of productization and product structure to enhance the software business perspective. The current state of productization and product structure practices were analyzed to understand the specific challenges related to the software industry and provide recommendations to address the challenges. The main identified challenge involves the essence of productization being missing in the software companies, and it seems that product-like objects are not being formed systematically, and the software product structure is not being considered from this perspective. Consequently, the logic for the software offering is lacking, the related documentation is lacking, and the terminology is not consistent. The study revealed special issues with software products and provided tangible contributions to align with the productization and product structure thinking from the software perspective. Adequate productization of software could support well-defined product variants and options, which would guide customers in their choice for configurations. The ineffective communication of customer requirements is one of the challenges brought by the lack of productization, hindering the final product execution by the software company. Three ways of productizing software business were proposed, including BPaaS, Own Premise, and SaaS. In addition, the product structure considerations were included. The proposed models may enhance software business management and the service business in the software context as they showcase a consistent logic in three types of software product offerings. Benefits are provided for both internal and external stakeholders.

Scientific Contributions

The study provides a unique contribution by considering productization and product structure in the software context and analyzing possibilities of systematizing software-related services. The findings support software product management efforts and selecting the requirements (Barney et al., 2008) by proposing certain systematics for managing the structure and ensuring consistency. The study also complements the present literature on productization (Harkonen et al., 2017) by involving a software example. Productization of software offering enhances the systematic, tangible, and formalized approach to facilitate understanding and managing the software product portfolios. This study specifically extends the understanding of software productization and takes the software business perspective. This study also contributes by addressing the standardization of software products through the product structure lens and hence supports the efforts towards standardized products Artz et al. (2010) and the shift away from customer specifics utilized for single customers only. Software portfolio considerations are encouraged in case of customer specifics. The automated variant customization (H. Zhang & Jarzabek, 2004) is supported by the proposed SaaS product structure providing the needed elements. Also, tackling the complex customer requirements in the software business (Cusumano, 2008) is supported by the software service productization providing certain standardization. Improving the product data management of the software business is also promoted by the product structure, providing increased consistency.

Active software product portfolio management is enabled by considering the commercial and technical productization perspectives in line with Tolonen et al. (2014). The commercial aspects are available and visible to the customers to be ordered and invoiced by the company. Product configurability is enabled through the predefined sales items, an aspect complementing Hanna, Harri, Olli, and Kongkiti (2011) and ElMaraghy et al. (2013) but providing new through the specific software perspective. This study also contributes to addressing concerns of configurability problems in the software business due to the complex nature of the software product business (Asikainen et al., 2003) by indicating the role of defined sales items. In the proposed SaaS productization, each software sales item is connected to their version items and sub-processes, cost drivers, and their determined needed resources on the technical side. The technical productization of the SaaS product structure is critical for the internal stakeholders such as R&D, software architects, and other software programmers.

Managerial Implications

This study has significant managerial implications that concern the software industry as it provides an effective tool for the management for considering products in the software business. Achieving long-term viability is generally critical for companies to survive, and therefore managers in the software business also have the responsibility to ensure the desirability and profitability of their products. The product structure can organize the seemingly complex software product portfolios into a logical and understandable form and provide clarity and consistency for the internal and external stakeholders. This study contributes by creating a platform for managers to know better what they are selling and understand the resources needed to create the software products. Commercial software productization can help to provide a common language regarding the products when communicating to customers. The software productization also helps to organize the software product portfolio for ease of documentation. It also improves the link and the roles between the technical and the commercial product structures. Software architects, software programmers, and the rest of R&D through the commercial and technical structure can now connect what constitutes software product sales items and their technical roles and responsibilities. Productization enables the tangibilisation and standardization of the service products in the software context, especially in the case of SaaS.

Limitations and Future Research

One potential limitation of this study is that results are based on empirical data from the analysis of one case company. This improved the possibilities of access to detailed information, but undoubtedly much more evidence is necessary from multiple companies to further confirm the findings. Another limitation relates to the nature of the studied company, not involving software companies characterized to provide packaged software, tailored software, or COTS software. Hence, it is necessary to repeat the study in software companies of other nature to have a wider representation of the software business. For example, future studies could analyze the impact of the inherent nature of the software business of providing new releases on productization. Also, the influence of agile methodologies such as Scrum would be an interesting perspective in the context. Future researchers could also investigate software product data management and productization in the wider scope of product portfolio management.

CONCLUSION

This study first addresses the productization concept and related product structure in the software business context. Then, three different forms of software product delivery are considered, including software as a service, own premise package, and business process as a service. This type of approach enables consistency in the provision of software, alongside certain systematics, and a level of standardization. Furthermore, effective productization enables dealing with both the commercial interface relating to the software products and that of the technical realization. The software products and services can become more manageable as a result.

ACKNOWLEDGEMENT

Foundation for Economic Education supported this work: [Grant Number 18-10307]

REFERENCES

Alajoutsijärvi, K., Mannermaa, K., & Tikkanen, H. (2000). Customer relationships and the small software firm: A framework for understanding challenges faced in marketing. *Information & Management*, *37*(3), 153–159. doi:https://doi.org/10.1016/S0378-7206(99)00039-7

Aldrich, J., Chambers, C., & Notkin, D. (2002). Archjava: Connecting software architecture to implementation. In *Proceedings of the 24th International Conference on Software Engineering,* Orlando, FL (pp. 187–197). doi:https://doi.org/10.1109/ICSE.2002.1007967

Alves, C., Pereira, S., & Castro, J. (2006). A study in market-driven requirements engineering. Retrieved from https://bit.ly/3h5tCcZ

Alves, V., Niu, N., Alves, C., & Valença, G. (2010). Requirements engineering for software product lines: A systematic literature review. *Information and Software Technology*, *52*(8), 806–820. doi:https://doi.org/10.1016/j.infsof.2010.03.014

Artz, P., Van De Weerd, I., & Brinkkemper, S. (2010). *Productization: The process of transforming from customer-specific software development to product software development* (Tech. Rep.). Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands.

Asagba, P. O., & Ogheneovo, E. E. (2008). A comparative analysis of structured and object-oriented programming methods. *Journal of Applied Sciences and Environmental Management*, *12*(4), 41-46. doi:https://doi.org/10.4314/jasem.v12i4.55217

Asikainen, T., Soininen, T., & Männistö, T. (2003). A koala-based approach for modelling and deploying configurable software product families. In *International Workshop on Software Product-Family Engineering,* Springer, Berlin, Heidelberg (pp. 225–249). doi:https://doi.org/10.1007/978-3-540-24667-1_17

Barney, S., Aurum, A., & Wohlin, C. (2008). A product management challenge: Creating software product value through requirements selection. *Journal of Systems Architecture*, *54*(6), 576–593. doi:https://doi.org/10.1016/j.sysarc.2007.12.004

Bhardwaj, S., Jain, L., & Jain, S. (2010). An approach for investigating perspective of cloud Software-as-a-Service (SaaS). *International Journal of Computer Applications*, *10*(2), 40–43. doi:https://doi.org/10.5120/1450-1962

Biernacki, P., & Waldorf, D. (1981). Snowball sampling: Problems and techniques of chain referral sampling. *Sociological Methods & Research*, *10*(2), 141–163. doi:https://doi.org/10.1177/004912418101000205

Choudhary, V. (2007). Software as a service: Implications for investment in software development. In *40th Annual Hawaii International Conference on System Sciences (HICSS'07),* Waikoloa, HI (pp. 209a–209a). doi:https://doi.org/10.1109/HICSS.2007.493

Cockburn, A. (2004). *Crystal clear: A human-powered methodology for small teams*. London, UK: Pearson Education.

Cooper, R., Edgett, S., & Kleinschmidt, E. (2001). Portfolio management for new product development: Results of an industry practices study. *r&D Management*, *31*(4), 361–380. doi:https://doi.org/10.1111/1467-9310.00225

Crnkovic, I., Asklund, U., & Dahlqvist, A. P. (2003). *Implementing and integrating product data management and software configuration management*. Boston, MA: Artech House.

Cusumano, M. A. (2008). The changing software business: Moving from products to services. *Computer*, *41*(1), 20–27. doi:https://doi.org/10.1109/MC.2008.29

Cusumano, M. A. (2019). 'platformizing'a bad business does not make it a good business. *Communications of the ACM*, *63*(1), 23–25. doi:https://doi.org/10.1145/3372918

Dang, S., Amin, O., Shihada, B., & Alouini, M.-S. (2020). What should 6g be? *Nature Electronics*, *3*(1), 20–29. doi:https://doi.org/10.1038/s41928-019-0355-6

Dubey, A., & Wagle, D. (2007). Delivering software as a service. *The McKinsey Quarterly*, *6*, 1-12.

ElMaraghy, H., Schuh, G., ElMaraghy, W., Piller, F., Schönsleben, P., Tseng, M., & Bernard, A. (2013). Product variety management. *Cirp Annals*, *62*(2), 629–652. doi:https://doi.org/10.1016/j.cirp.2013.05.007

Feller, J., Finnegan, P., Fitzgerald, B., & Hayes, J. (2008). From peer production to productization: A study of socially enabled business exchanges in open source service networks. *Information Systems Research*, *19*(4), 475–493. doi:https://doi.org/10.1287/isre.1080.0207

Flamholtz, E. (1995). Managing organizational transitions: Implications for corporate and human resource management. *European Management Journal*, *13*(1), 39–51. doi:https://doi.org/10.1016/0263-2373(94)00056-D

Fricker, S., & Kittlaus, H.-B. (2017). *Software product management: The ispma-compliant study guide and handbook*. Berlin, Heidelberg: Springer-Verlag.

Fricker, S. A. (2012). Software product management. In *Software for people* (p. 53-81). Berlin, Heidelberg: Springer.

Gbadegeshin, S. A. (2019). The effect of digitalization on the commercialization process of high-technology companies in the life sciences industry. *Technology Innovation Management Review*, *9*(1), 49-63. doi:https://doi.org/10.22215/timreview/1211

Gharbi, M., Koschel, A., & Rausch, A. (2019). *Software architecture fundamentals: A study guide for the certified professional for software architecture®–foundation level–isaqb compliant*. Heidelberg, Germany: Dpunkt. verlag.

Godse, M., & Mulik, S. (2009). An approach for selecting Software-as-a-Service (SaaS) product. In *IEEE International Conference on Cloud Computing,* Bangalore, India (pp. 155–158). doi:https://doi.org/10.1109/CLOUD.2009.74

Haines, S. (2014). *Product manager's desk reference*. New York, NY: McGraw-Hill Education.

Hanna, K.-V., Harri, H., Olli, J., & Kongkiti, P. (2011). Product configuration management in ict companies: The practitioners' perspective. *Technology and Investment*, *2*, 273-285. doi:https://doi.org/10.4236/ti.2011.24028

Hannila, H., Tolonen, A., Harkonen, J., & Haapasalo, H. (2019). Product and supply chain related data, processes and information systems for product portfolio management. *International Journal of Product Lifecycle Management*, *12*(1), 1–19. doi:https://doi.org/10.1504/IJPLM.2019.104352

Harkonen, J., Haapasalo, H., & Hanninen, K. (2015). Productisation: A review and research agenda. *International Journal of Production Economics*, *164*, 65–82. doi:https://doi.org/10.1016/j.ijpe.2015.02.024

Harkonen, J., Mustonen, E., & Hannila, H. (2019). Productization and product structure as the backbone for product data and fact-based analysis of company products. In *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM),* Macao, China (pp. 474–478). doi:https://doi.org/10.1109/IEEM44572.2019.8978845

Harkonen, J., Tolonen, A., & Haapasalo, H. (2017). Service productisation: Systematising and defining an offering. *Journal of Service Management*, *28*(5), 936-971. doi:https://doi.org/10.1108/JOSM-09-2016-0263

Harkonen, J., Tolonen, A., & Haapasalo, H. (2018a). Modelling of manufacturing services and processes for effective productization. In *20th International Working Seminar on Production Economics,* Innsbruck, Austria (p. 19-23).

Harkonen, J., Tolonen, A., & Haapasalo, H. (2018b). Modelling of construction products and services for effective productisation. *Management (18544223)*, *13*(4), 335-353.

Helferich, A., Schmid, K., & Herzwurm, G. (2006). Product management for software product lines: An unsolved problem? *Communications of the ACM*, *49*(12), 66–67. doi:https://doi.org/10.1145/1183236.1183268

Hietala, J., Kontio, J., Jokinen, J.-P., & Pyysiainen, J. (2004). Challenges of software product companies: Results of a national survey in Finland. In *10th International Symposium on Software Metrics,* Chicago, IL (pp. 232–243). doi:https://doi.org/10.1109/METRIC.2004.1357906

Hinings, B., Gegenhuber, T., & Greenwood, R. (2018). Digital innovation and transformation: An institutional perspective. *Information and Organization*, *28*(1), 52–61. doi:https://doi.org/10.1016/j.infoandorg.2018.02.004

Jaakkola, E. (2011). Unraveling the practices of "productization" in professional service firms. *Scandinavian Journal of Management*, *27*(2), 221–230. doi:https://doi.org/10.1016/j.scaman.2011.03.001

Jayasimha, K., Nargundkar, R. V., et al. (2020). Impact of Software as a Service (SaaS) on software acquisition process. *Journal of Business & Industrial Marketing*, *35*(4), 757-770. doi:https://doi.org/10.1108/JBIM-12-2018-0382

Kahn, K. B. (2012). *The PDMA handbook of new product development*. New York, NY: John Wiley & Sons.

Kasanen, E., Lukka, K., & Siitonen, A. (1993). The constructive approach in management accounting research. *Journal of Management Accounting Research*, *5*(1), 243–264.

Kästner, C., Apel, S., & Ostermann, K. (2011). The road to feature modularity? In *Proceedings of the 15th International Software Product Line Conference,* New York, NY (pp. 1–8). doi:https://doi.org/10.1145/2019136.2019142

Kim, K., Kim, H., & Kim, W. (2007). Building software product line from the legacy systems" experience in the digital audio and video domain". In *11th International Software Product Line Conference (SPLC 2007),* Kyoto, Japan (pp. 171–180). doi:https://doi.org/10.1109/SPLINE.2007.27

Kotler, P., & Keller, K. (2012). *Marketing management.* New Jersey, NJ: Prentice Hall.

Kropsu-Vehkaperä, H. (2012). *Enhancing understanding of company-wide product data management in ICT companies* (Vol. 418). Finland: Acta Universitatis Ouluensis, C Technica.

Kumar, G., & Bhatia, P. K. (2012). Impact of agile methodology on software development process. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, *2*(4), 46–50.

Kuula, S., Haapasalo, H., & Tolonen, A. (2018). Cost-efficient co-creation of knowledge intensive business services. *Service Business*, *12*(4), 779–808. doi:https://doi.org/10.1007/s11628-018-0380-y

Lahtinen, N., Mustonen, E., & Harkonen, J. (2019). Commercial and technical productization for fact-based product portfolio management over lifecycle (just accepted). *IEEE Transactions on Engineering Management*, 1-13. doi:https://doi.org/10.1109/TEM.2019.2932974

Light, B., & Papazafeiropoulou, A. (2004). Reasons behind ERP package adoption: A diffusion of innovations perspective. In *12th European Conference on Information Systems,* Turku, Finland.

Mathur, S. K. (2006). Indian information technology industry: Past, present and future and a tool for national development. *Journal of Theoretical and Applied Information Technology*, *2*(2), 50–79.

McKay, A., Erens, F., & Bloor, M. S. (1996). Relating product definition and product variety. *Research in Engineering Design*, *8*(2), 63–80. doi:https://doi.org/10.1007/BF01607862

MCSI, M. (2009). Configurability in SaaS (Software as a Service) applications. In *Proceedings of 2nd Annual Conference on India Software Engineering Conference,* New York, NY (p. 19-26). doi:https://doi.org/10.1145/1506216.1506221

Merton, R., Fiske, M., & Kendall, P. (1990). *The focused interview: A manual of problems and procedures*. New York, NY: The Free Press.

Meyer, M. H., & Webb, P. H. (2005). Modular, layered architecture: The necessary foundation for effective mass customisation in software. *International Journal of Mass Customisation*, *1*(1), 14–36. doi:https://doi.org/10.1504/IJMASSC.2005.007349

Mustonen, E., Harkonen, J., & Haapasalo, H. (2019). From product to service business: Productization of product-oriented, use-oriented, and result-oriented business. In *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM),* Macao, China (pp. 985–989). doi:https://doi.org/10.1109/IEEM44572.2019.8978581

Ostermann, K., Giarrusso, P. G., Kästner, C., & Rendel, T. (2011). Revisiting information hiding: Reflections on classical and nonclassical modularity. In *European conference on object-oriented programming,* Springer, Berlin, Heidelberg (pp. 155–178). doi:https://doi.org/10.1007/978-3-642-22655-7_8

Pandla, K. (2016). Drivers and characteristics of high performing organizations. *International Journal of Business and Administrative Studies*, *2*(3), 57–61. doi:https://doi.org/https://10.20469/ijbas.2.10001-3

Parry, S., Jones, R., Rowley, J., & Kupiec-Teahan, B. (2012). Marketing for survival: A comparative case study of SME software firms. *Journal of Small Business and Enterprise Development*, *19*(4), 712-728. doi:https://doi.org/10.1108/14626001211277488

Peltonen, H. (2000). *Concepts and an implementation for product data management*. Espoo, Finland: Finnish Academy of Technology.

Pyron, C., Prado, J., & Golab, J. (1998). Test strategy for the PowerPC 750 microprocessor. *IEEE Design & Test of Computers*, *15*(3), 90–97. doi:https://doi.org/10.1109/54.706039

Ratnayake, R. C., & Markeset, T. (2010). Implementing company policies in plant level asset operations: Measuring organisational alignment. *European Journal of Industrial Engineering*, *4*(3), 355–371. doi:https://doi.org/10.1504/EJIE.2010.033335

Rautiainen, K., Vuornos, L., & Lassenius, C. (2003). An experience in combining flexibility and control in a small company's software product development process. In *International Symposium on Empirical Software Engineering,* Rome, Italy (pp. 28–37). doi:https://doi.org/10.1504/10.1109/ISESE.2003.1237962

Saaksvuori, A., & Immonen, A. (2008). *Product lifecycle management*. Berlin, Germany: Springer Science & Business Media.

Sabin, D., & Weigel, R. (1998). Product configuration frameworks-a survey. *IEEE Intelligent Systems and their applications*, *13*(4), 42–49. doi:https://doi.org/10.1504/10.1109/5254.708432

Sakti, I. W. (2016). The analysis factors of experential marketing, product quality, and customer satisfaction of motor bike as a main transportation mode in Bandung-Indonesia. *International Journal of Business and Administrative Studies*, *2*(1), 6–8.

Salvador, F., Forza, C., & Rungtusanatham, M. (2002). Modularity, product variety, production volume, and component sourcing: Theorizing beyond generic prescriptions. *Journal of operations management*, *20*(5), 549–575. doi:https://doi.org/10.1016/S0272-6963(02)00027-X

Saunders, M., Lewis, P., & Thornhill, A. (2007). *Research methods for business students.* Hoboken, NJ: Prentice Hall.

Sawyer, S. (2000). Packaged software: Implications of the differences from custom approaches to software development. *European Journal of Information Systems*, *9*(1), 47–58. doi:https://doi.org/10.1057/palgrave.ejis.3000345

Schwaber, K., & Beedle, M. (2002). *Agile software development with scrum.* Upper Saddle River, NJ: Prentice Hall.

Solms, F. (2012). What is software architecture? In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference,* New York, NY (p. 363-373). doi:https://doi.org/10.1145/2389836.2389879

Srinivasan, S. R., Ramakrishnan, S., & Grasman, S. E. (2005). Identifying the effects of cannibalization on the product portfolio. *Marketing Intelligence & Planning*, *23*(4), 359-371. doi:https://doi.org/10.1108/02634500510603465

Stark, J. (2016). Product lifecycle management. In *Product lifecycle management.* Cham, Switzerland: Springer.

Subair, S. (2014). The evolution of software process models: From the waterfall model to the Unified Modelling Language (UML). *International Journal of Information & Technology Systems*, *3*(2), 7–14.

Sudarsan, R., Fenves, S. J., Sriram, R. D., & Wang, F. (2005). A product information modeling framework for product lifecycle management. *Computer-Aided Design*, *37*(13), 1399–1411. doi:https://doi.org/10.1016/j.cad.2005.02.010

Suominen, A., Kantola, J., & Tuominen, A. (2009). Reviewing and defining productization. In *20th Annual Conference of the International Society for Professional Innovation Management (ISPIM 2009),* Vienna, Austria.

Svensson, D., & Malmqvist, J. (2002). Strategies for product structure management at manufacturing firms. *Journal of Computing and Information Science in Engineering*, *2*(1), 50–58. doi:https://doi.org/https://10.1115/1.1471356

Tolonen, A., Harkonen, J., Haapasalo, H., et al. (2014). Product portfolio management—governance for commercial and technical portfolios over life cycle. *Technology and investment*, *5*(04), 173-183. doi:https://doi.org/10.4236/ti.2014.54016

Tolonen, A., Harkonen, J., Verkasalo, M., & Haapasalo, H. (2015a). Product portfolio management process over horizontal and vertical portfolios. *International Journal of Product Lifecycle Management*, *8*(3), 189–215. doi:https://doi.org/10.1504/IJPLM.2015.074132

Ulrich, K., & Eppinger, S. (2012). *Product design and development.* New York, NY: McGraw-Hill.

Vlaanderen, K., Brinkkemper, S., & van de Weerd, I. (2012). On the design of a knowledge management system for incremental process improvement for software product management. *International Journal of Information System Modeling and Design (IJISMD)*, *3*(4), 46–66. doi:https://doi.org/10.4018/jismd.2012100103

Wallin, C., Ekdahl, F., & Larsson, S. (2002). Integrating business and software development models. *IEEE Software*, *19*(6), 28–33. doi:https://doi.org/10.1109/MS.2002.1049384

Wu, L., Garg, S. K., & Buyya, R. (2011). SLA-based resource allocation for Software as a Service provider (SaaS) in cloud computing environments. In *11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing,* Newport Beach, CA (pp. 195–204). doi:https://doi.org/10.1109/CCGrid.2011.51

Xu, L., & Brinkkemper, S. (2007). Concepts of product software. *European Journal of Information Systems*, *16*(5), 531–541. doi:https://doi.org/10.1057/palgrave.ejis.3000703

Youngdahl, W. E., Ramaswamy, K., & Dash, K. C. (2010). Service offshoring: The evolution of offshore operations. *International Journal of Operations & Production Management*, *30*(8), 798-820. doi:https://doi.org/10.1108/01443571011068171

Zhang, H., & Jarzabek, S. (2004). XVCL: A mechanism for handling variants in software product lines. *Science of Computer Programming*, *53*(3), 381–407. doi:https://doi.org/10.1016/j.scico.2003.04.007

Zhang, S., Shen, W., & Ghenniwa, H. (2004). A review of internet-based product information sharing and visualization. *Computers in Industry*, *54*(1), 1–15. doi:https://doi.org/10.1016/j.compind.2003.09.002